MASTER EN SCIENCES INFORMATIQUES

Une approche méthodologique et un outil pour la construction de cahiers des charges pour la sélection de logiciels

Capelle, David; Grimard, Laurent

Award date:
2003

Facultés Universitaires Notre-Dame de la Paix, Namur
Institut d'Informatique
Année académique 2002-2003

«Une approche méthodologique et un outil pour la construction de cahiers des charges pour la sélection de logiciels»

David Capelle Laurent Grimard

Mémoire présenté en vue de l'obtention du grade de Maître en informatique.

Résumé

Ce document présente une approche méthodologique et un outil pour la construction du cahier des charges destiné à la sélection d'un logiciel. Le domaine de ce travail se limite aux exigences non fonctionnelles avec une attention particulière portée à leurs aspects sécurité.

Ce document se décline en trois grandes parties. La première est un état de l'art en matière d'exigences non fonctionnelles. Celles-ci ont été adaptées pour l'acquisition logicielle et réexprimées sous forme d'exigences types pour disposer d'un template instanciable. La seconde partie présente une méthodologie pour la sélection des exigences types. L'instanciation des exigences types est illustrée par un exemple pour le domaine des ERP. La troisième partie décrit les principales fonctionnalités et l'architecture du logiciel OPAL. Ce logiciel a pour objectif principal d'aider à la création d'un cahier des charges en permettant, par exemple, la réutilisation des cahiers des charges précédemment construits.

Mots Clés : cahier des charges, exigences non fonctionnelles, sécurité, sélection de logiciel, exigences types, méthodologie de sélection d'exigences, OPAL, ERP

Abstract

This document presents a methodological approach and a tool for the construction of the software requirements. The field of this work is limited to the non functional requirements with a particular attention paid to their security aspects.

This document is composed of three main parts. The first one is a state of the art concerning the non functional requirements. Those were adapted for software selection and re-expressed as a more generic form called 'standard requirement'. The second part presents a methodology for the selection of the standard requirements. The instanciation of those requirements is illustrated by an example (ERP). The third part describes the principal functionalities and the architecture of the OPAL software. The main objective of this software is to help the requirements management by allowing, for example, the re-use of previously built requirements.

Keywords : software requirements, non functional requirements, security, software selection, standard requirements, methodology of requirements selection, OPAL, ERP

Nous tenons tout d'abord à remercier Monsieur Eric Dubois, notre promoteur, qui nous a aidés et guidés durant la réalisation de ce mémoire. Nous le remercions tout particulièrement pour son dévouement et sa compréhension sans lesquels ce mémoire n'aurait pas été possible.

Nous tenons également à montrer notre gratitude à Monsieur Brice Bucciarelli (Centre de Recherche Publique Henri Tudor) qui nous a consacré une part de son précieux temps et nous a donné de nombreux conseils afin d'améliorer la qualité de notre travail.

Nous remercions également l'ensemble des membres du CITI (Centre de recherche Henri Tudor) et plus particulièrement Messieurs Marc Krystkowiak et Stefan Leidner ainsi que Madame Céline Décosse pour leur aide dans notre recherche de documentation.

Table des matières

Table des figures

Glossaire

Activité métier : Activité liée à un type de métier. Les différentes exigences du cahier des charges peuvent être liées à une ou à un ensemble d'activités métiers afin d'assurer un certain niveau de traçabilité.

Attribut : Elément constitutif ou Caractéristique d'une donnée, d'un utilisateur ou d'une fonctionnalité. Il existe deux types d'attributs, les attributs qui contiennent des informations ayant un simple but informatif (par exemple le nom d'un fichier ou d'un utilisateur). D'autres, comme c'est le cas des attributs de sécurité, déterminent un comportement du système (par exemple, les attributs de droits d'accès). Par exemple, les attributs d'un utilisateur sont son nom, prénom, rôle, etc.

Biométrie : Etude des éléments physiologiques uniques et propres à chaque individu (Rétine, empreinte digitale, etc.)

Business domain : Catégorie précise de logiciel (par exemple CRM, ERP, etc.) qui est utilisée afin de catégoriser les projets et les rendre ainsi plus facilement accessibles.

Catégorie : Catégorie-packaging d'éléments du cahier des charges (par exemple, IHM). La catégorisation d'une exigence permet d'assurer un certain niveau de traçabilité des exigences.

Checksum : Mécanisme permettant de vérifier l'intégrité d'un fichier en vérifiant l'égalité entre la somme des bits du fichier (calculée à la lecture) et la somme inscrite dans le fichier lors de sa création.

Chevaux de Troie : Programme informatique effectuant des opérations malicieuses à l'insu de l'utilisateur (la plupart du temps en donnant accès à des fonctions du système infecté à des utilisateurs extérieurs).

Concepts Avancés : Ensemble reprenant les activités métiers, les catégories et les goals d'un projet.

Contrôle du flux d'information : Contrôle de l'information qui entre et sort de l'entreprise. Ce contrôle permet la détection de virus, de chevaux de Troie, de spams, des informations confidentielles, etc.

Cryptage asymétrique : Le cryptage asymétrique consiste à utiliser des clés différentes pour le cryptage et le décryptage. Chaque correspondant diffuse une clé publique et conserve sa clé privée. Un message peut être crypté avec la clé publique du destinataire afin que seul celui-ci puisse le lire (avec sa clé privée).

Degré de partage d'un projet : Le degré de partage d'un projet reflète les droits d'accès des autres utilisateurs à ce projet.

Données : Informations utilisées par un logiciel. Elles peuvent être créées par l'utilisateur ou par le programme lui-même.

Données sensibles : Données nécessaires à l'activité de l'entreprise

Evènement d'audit : Evènement qu'il est possible d'auditer (par exemple un accès à une donnée).

Evènement à auditer : Evénement d'audit qui sera analysé.

Fonctionnalité critique : Fonctionnalité impérative à l'activité de l'entreprise.

GMAO : Gestion de Maintenance Assistée par Ordinateur

Goal : But ou sous-but d'un projet. Les goals sont utilisés pour classifier les exigences d'un cahier des charges afin d'assurer un certain niveau de traçabilité des exigences. Un goal peut être par exemple 'augmenter la compétitivité de l'entreprise'.

Information résiduelle : Donnée informatique étant encore physiquement présente sur un support alors qu'elle a été, au préalable, supprimée logiquement.

Maintenance : ensemble des tâches ayant pour objectif l'entretien d'un système.

MAPI : (Anglais:Messaging Application Programming Interface) API de messagerie de Microsoft.

Modèle de préférences : Ensemble rassemblant un système de pondération, un système de notation, un glossaire ainsi qu'un ensemble de structures de description (une pour chaque partie du cahier des charges).

MTBF : *(Anglais : Mean Time Between Failure)* Temps moyen estimé par statistique entre deux pannes.

MTTR : (*Anglais : Mean Time To Repair*) Temps moyen estimé pour une réparation

ODBC : Open Data Base Connection, standard de middelware de Microsoft pour accéder à des bases de données serveur à partir d'un micro ordinateur (client ou serveur)

OLE: Object Linking and Embedding, est un standard de partage et d'échange de données. Technique mise au point par Microsoft pour inclure dans un document, des documents d'autres applications, selon le principe du « client/serveur », i.e. en gardant le lien avec l'application d'origine, qu'on pourra rappeler pour une modification ou une mise à jour.

OPAL : Outillage du Processus d'Acquisition d'un Logiciel

Réseau filaire : Réseau dont la transmission de données se fait par fil. Les terminaux sont donc fixes et sont reliés par des câbles électriques ou optiques.

Réseau sans fil WIFI : (Anglais : Wireless Fidelity) Réseau qui utilise des fréquences hertziennes pour ses transmissions.

Signature électronique : Procédé permettant de garantir l'authenticité de l'expéditeur (fonction d'authentification), ainsi que de vérifier l'intégrité du message reçu.

Structures de description d'une exigence : Structure via laquelle une exigence va être décrite. Dans OPAL, il est possible de définir la structure de description utilisée pour chaque partie du cahier des charges (Présentation, exigences fonctionnelles, exigences non fonctionnelles, critères d'appel d'offres).

Système de notation : Ce système permet de quantifier la réponse d'un fournisseur par rapport à une exigence. Par exemple, exigence totalement satisfaite, partiellement satisfaite ou non satisfaite.

Système de pondération : Ensemble des pondérations possibles d'une exigence (par exemple, accessoire, importante, très importante). Ce système va donc permettre de gérer l'importance relative des exigences entre elles.

Système embarqué : Appareil spécialisé conçu pour exécuter des tâches spécifiques au sein d'une entreprise. L'exemple le plus courant est l'assistant personnel aussi nommé PDA (Personal Digital Assistant)

Template : Document générique pouvant servir de base à un travail.

Terminology Set : Ensemble reprenant la 'traduction' d'un certain nombre de keywords afin d'être affichée dans l'IHM d'OPAL. Le keyword EXF peut par exemple être associé au terme 'exigences fonctionnelles' afin que ce terme soit affiché dans un onglet.

Spam : Message électronique non sollicité.

Virus : Programme informatique hostile pouvant mettre en danger un système informatique.

Introduction générale

De nos jours, la place occupée par l'informatique dans notre société est de plus en plus importante. De plus, la complexité des programmes informatiques est sans cesse croissante. C'est pourquoi une spécification correcte des besoins lors de l'acquisition ou la création de logiciels est essentielle. Ces spécifications sont formalisées dans un document appelé 'Cahier Des Charges'.

Lors de la création d'un tel document pour un logiciel, il est nécessaire de spécifier l'ensemble des exigences auxquelles celui-ci devra répondre. Les spécifications peuvent être divisées en deux catégories majeures :

- Les spécifications fonctionnelles qui précisent les traitements à effectuer, le comportement exigé.
- Les spécifications autres que le comportement, dites spécifications non-fonctionnelles. Ce type d'exigences est très important car les fonctionnalités seules ne constituent pas l'intégralité des besoins de l'utilisateur.

En outre, de plus en plus de sociétés n'ont pas la capacité nécessaire à la création des logiciels complexes, nécessaires à leur fonctionnement et à leur compétitivité. Ces sociétés se tournent donc vers l'acquisition d'un logiciel déjà produit et qui sera paramétré selon leurs besoins.

Dans ce mémoire, nous nous focalisons donc sur les spécifications à fournir lors de l'**acquisition** d'un logiciel. En effet, à ce jour, aucune méthodologie n'a été conçue dans ce but. De plus, nous nous contentons d'étudier les **exigences non fonctionnelles** (et plus spécialement les exigences concernant la sécurité) car dans le domaine de l'acquisition logicielle, les différents produits disponibles sur le marché proposent souvent le même type de fonctionnalités.

L'objectif de ce mémoire est multiple :

1. Proposer un état de l'art en matière d'exigences non fonctionnelles, avec une attention particulière envers les exigences de sécurité ;
2. Mettre en avant l'aspect systématique de la définition de ces exigences en leurs donnant un aspect générique et instanciable ;
3. Proposer une méthodologie capable d'automatiser en partie la sélection des exigences et montrer comment les exigences génériques peuvent être instanciées selon un contexte ;
4. Fournir un outil logiciel capable d'aider à la construction d'un cahier des charges.

Le premier chapitre propose un état de l'art en matière d'exigences non fonctionnelles. Ces exigences proviennent de plusieurs templates ou normes (Template Volere, IEEE 830 et ISO 15408). Elles ont parfois été légèrement modifiées afin d'être compatibles avec le domaine de l'acquisition logicielle. De plus, les exigences sont présentées sous forme d''exigences types', c'est-à-dire sous une forme permettant une instanciation lors de sa mise en contexte.

Les sections du premier chapitre décrivent les grands types d'exigences présentes dans un cahier des charges (sécurité, performance, maintenance, etc.). Chaque section est ensuite développée jusqu'au niveau des 'exigences types'. Quelques exemples d'instanciation sont également indiqués afin de permettre une meilleure perception de l'exigence.

Le deuxième chapitre est consacré à la mise en pratique des exigences définies dans le premier chapitre et d'en montrer l'utilité. Pour ce faire, nous avons adopté une méthodologie qui consiste, pour un domaine précis (dans notre exemple, les ERP), à spécifier pour chaque exigence type (ou ensemble d'exigences) les critères qui définiront si cette exigence doit se trouver ou non dans le cahier des charges. Le nombre d'utilisateurs est un exemple de critère : il peut être élevé, moyen ou faible.

Ce chapitre étant une mise en pratique du premier chapitre, leurs structures sont donc tout à fait identiques.

Le résultat de cette méthodologie fournit une nouvelle structure qui, une fois adaptée à un cas réel (où on peut donner une valeur réelle au critère, par exemple 'le nombre d'utilisateur est élevé'), permet de sélectionner aisément les exigences devant se trouver dans le cahier des charges à réaliser. Cette méthodologie permet donc d'automatiser en partie la sélection des exigences ce qui représente un gain de temps considérable lors de la création d'un cahier des charges.

Le dernier chapitre quant à lui est consacré à un autre type d'aide à l'expression des exigences. Ainsi, nous y présentons le logiciel OPAL que nous avons aidé à développer lors de notre stage. OPAL est un logiciel d'aide à la création de cahiers des charges qui permet également d'animer l'appel d'offres. OPAL n'a évidemment pas la prétention de se substituer à d'autres logiciels du marché mais apporte néanmoins une aide en permettant la réutilisation de projets précédents.

Nous présentons dans ce chapitre les principales fonctionnalités du logiciel ainsi que quelques éléments de son architecture.

Chapitre 1 : Etat de l'art en matière d'exigences non fonctionnelles

Introduction

Ce chapitre contient un état de l'art des exigences **non fonctionnelles** pour lesquelles de nombreuses normes sont à la disposition du consultant afin de l'aider dans la rédaction des ces exigences. Ces normes sont parfois très différentes dans leur structure et leur contenu. Il est donc utile d'avoir à disposition un document reprenant une synthèse, sous un format unique, de toutes ces normes et templates.

Dans ce chapitre, nous avons également sélectionné ou adapté les exigences retenues au domaine plus précis de l'**acquisition logicielle** car les logiciels sont de plus en plus 'commandés' ou 'adaptés' en fonction de besoins spécifiques. En effet, les logiciels actuels étant de plus en plus complexes, peu de sociétés peuvent se permettre de les développer en interne.

Nous avons également décidé de porter une attention toute particulière aux exigences de **sécurité**. En effet, la sécurité devient de nos jours un point crucial pour de nombreuses applications et une mauvaise définition des besoins peut avoir des répercussions conséquentes.

Le but de ce chapitre est double :

- Réaliser un état de l'art en matière d'exigences non fonctionnelles en se basant sur plusieurs templates/normes.
- Redéfinir ces exigences en termes d' 'Exigences types'.

Par 'exigence type' nous entendons la redéfinition d'une exigence sous une forme plus générique. Cette forme permettra une instanciation lors de sa mise en contexte. Le chapitre suivant présente un exemple de la méthodologie de mise en contexte d'exigences types afin d'illustrer l'apport de cette méthode.

Les normes sur lesquelles nous avons basé notre réflexion sont le template Volere, le IEEE 830 et la norme ISO 15408. En voici une brève description :

Volere Requirements Specification Template Version 8 [VOL98]

> Ce template, dont les auteurs sont James et Suzanne Roberston, est publié par l'Atlantic Systems Guild[1], une organisation de consultance spécialisée dans la construction de systèmes. Cette guilde a pour but principal d'offrir des techniques pragmatiques à l'efficacité prouvée afin d'aider à l'élicitation et la spécification des exigences.
> C'est pour cette raison qu'un ensemble de ressources ont été publiées sous le nom de Volere[2]. Ces ressources s'appuient sur l'expérience de l'ingénierie des exigences dans de nombreux domaines d'application et sur une large palette de recherches académiques. Elles visent à fournir la base pour améliorer les spécifications d'exigences.
>
> Le template Volere est un cahier des charges type dont les buts principaux sont :
>
> - Fournir une structure adéquate à la description des exigences ;
> - Proposer une structure de cahier des charges complète et adaptable selon la situation.

[1] http://www.systemsguild.com
[2] http://www.volere.co.uk

Les points principaux du cahier des charges types sont :

- Les conducteurs du projet (but du produit, utilisateurs du produit,...)
- Les contraintes sur le projet (environnement technologique)
- Les conventions de nomage et définitions
- Les faits et hypothèses utiles (facteurs externes influençant le projet)
- Les exigences fonctionnelles (description des fonctionnalités demandées)
- Les exigences non fonctionnelles (performances,...)
- Les autres aspects du projet (tâches de développement, etc.)

Ce template (et plus spécialement la partie consacrée aux exigences non fonctionnelles) a servi de base à notre travail. Nous l'avons utilisé pour l'ensemble des points de ce chapitre, bien qu'il ait initialement été conçu pour le développement de logiciels.

IEEE Recommended Practice for Software Requirements Specifications (IEEE 830) [IEE98]

Cette norme, issue de l'organisme 'Institute of Electrical and Electronics Engineers' [3], a pour but de définir une liste de pratiques recommandées pour la spécification des exigences d'un programme.
Ce document est divisé en 2 grands thèmes :

- Une liste de points à prendre en considération lors de la création des exigences (caractéristiques d'un bon cahier des charges, le prototypage,...)
- Les parties principales d'une spécification correcte.

En voici sa structure :

- Introduction (but du projet, étendue du projet, etc.)
- Description générale (caractéristiques des utilisateurs, fonction du produit, etc.)
- Exigences spécifiques (performance, contrainte de design, etc.)
- Information de support (table des matières, annexe, etc.)

Dans ce chapitre, nous nous sommes basés sur la version de 1993 de cette norme (éditée pour la première fois en 1984).

Bien que la structure générale de notre travail ait été inspirée de la norme Volere, l'influence de l'IEEE 830 est présente à différents niveaux dans ce chapitre.

Common Criteria (ISO 15408) [CC98]

Les critères communs[4] (CC) représentent un aboutissement des efforts consentis pour développer des critères d'évaluation de la sécurité des technologies de l'information qui soient largement utilisables par la communauté internationale. Ils résultent de l'harmonisation et de l'amélioration d'un certain nombre de critères sources, soit les critères européens (ITSEC), les critères américains (TCSEC), et les critères canadiens (CTCPEC), et en éliminent les différences conceptuelles et techniques. Les critères communs ouvrent donc la voie à la reconnaissance des résultats d'évaluation dans le monde entier.

[3] www.ieee.com
[4] www.iso.com

Ces critères présentent les exigences relatives à la sécurité d'un produit ou d'une technologie sous deux formes distinctes : les exigences fonctionnelles et les exigences d'assurance. Les exigences fonctionnelles définissent le comportement de sécurité d'un produit. Les exigences d'assurance constituent la base de confiance déterminée lorsque les mesures de sécurité sont efficaces et conformes aux spécifications.

Le document sur lequel nous nous sommes appuyés, « partie 2 : Exigences de sécurité fonctionnelle », organise les exigences fonctionnelles hiérarchiquement : d'abord, les classes qui sont un groupement de famille d'exigences partageant le même objectif ; ensuite, les familles d'exigences qui sont un groupement de composants partageant les même objectifs mais pouvant différer dans l'accentuation et la rigueur et pour finir, les composants qui constituent le plus petit ensemble utilisable d'éléments qui peuvent être inclus dans un profil de protection ou une cible de sécurité. Il existe onze classes de sécurité (dont l'audit, l'identification et l'authentification, la protection de la vie privée, etc.).

Méthodologie :

- Les développeurs utilisent les exigences fonctionnelles des Common Criteria pour choisir le niveau de sécurité de leur produit.
- Les évaluateurs utilisent les exigences d'assurance pour indiquer le niveau de confiance que l'on a en l'implémentation qui a été faite des exigences fonctionnelles.
- Les utilisateurs connaissent la fiabilité du produit qu'ils utilisent grâce à sa documentation (description fonctionnelle) et le rapport d'évaluation (niveau d'assurance du produit).

Ce document a principalement été utilisé pour la section 'Sécurité' de ce chapitre

Vous trouverez ci-dessous l'état de l'art des exigences non fonctionnelles ainsi que leur mise sous forme d'exigences types.

Conventions de notation

Les crochets « [] » indiquent les parties facultatives. Les crochets suivis de parenthèse « [()] » indiquent les parties ne devant pas se retrouver dans l'instance d'exigence. Les éléments en italique représentent des exemples d'instances d'une exigence.

1.1. La sécurité du système

1.1.1. Contrôle d'accès au système

Cette partie spécifie l'ensemble des exigences relatives à l'établissement d'une session d'un utilisateur sur le système. Les exigences sur la gestion du contrôle d'accès au système doivent se trouver dans la partie 'gestion de la sécurité' à la section 'Gestion des fonctionnalités et données du système' (point 1.1.4.1.). Les exigences sur la gestion des comptes utilisateurs doivent se trouver dans la partie 'Gestion des comptes utilisateurs' (point 1.1.4.2.).

1.1.1.1. Limitation du domaine aux attributs de sécurité de la session

Ce point définit les exigences garantissant que lors d'une ouverture de session, les attributs de sécurité de celle-ci seront limités aux droits et privilèges (définis par les attributs de sécurité) de l'utilisateur. L'accès sera dès lors déterminé par les attributs de la session de l'utilisateur et non plus par l'utilisateur lui-même.
Dans la suite de ce document, nous continuerons, pour simplifier, à utiliser l'expression « attributs de sécurité des utilisateurs », ceci afin de faciliter la lecture et la compréhension.

Exigence type

Le système doit limiter le domaine des attributs de sécurité de la session d'un utilisateur en fonction des attributs de sécurité $X_{1..N}$. de celui-ci.

Où X est un attribut de sécurité d'un utilisateur (identité, groupe, rôle, niveau de sécurité ou d'intégrité, etc.).

- Le système doit limiter le domaine des attributs de sécurité de la session d'un utilisateur en fonction des *droits d'accès aux fonctionnalités et données* de celui-ci.

1.1.1.2. Etablissement d'une session

Ce point décrit l'exigence pour refuser à un utilisateur la permission d'établir une session avec le système.

Exigence type

Le système doit être capable de refuser l'ouverture d'une session en fonction d'un ensemble $X_{1..N}$ d'attributs.

Où X est un attribut d'utilisateur (nom, prénom, login, mot de passe, droits d'accès, etc.).

- Le logiciel doit être capable de refuser l'ouverture d'une session *si l'utilisateur se trompe de mot de passe.*

1.1.1.3. *Limitation du nombre de sessions parallèles*

Ce point définit l'exigence permettant de mettre des limites au nombre de sessions parallèles qu'un même utilisateur peut avoir.

Exigence type

Le système doit pouvoir gérer un nombre N de sessions parallèles pour un même utilisateur.

1.1.1.4. *Verrouillage de session*

Exigence type

Dans des circonstances $X_{1..N}$, le système doit verrouiller la session de l'utilisateur en effectuant une série $Y_{1..N}$ de mesure rendant l'accès aux données de l'utilisateur impossible sans un déverrouillage.

Où X est une circonstance (après un temps d'inactivité, demande de l'utilisateur, etc.).
Y est une action effectuée par le système (effacer, écraser le contenu de l'image, demander un déverrouillage, etc.).

- *Après un certain temps d'inactivité*, le logiciel doit verrouiller la session de l'utilisateur *en effaçant le contenu de l'écran d'affichage et en désactivant tout moyen d'accès à cette session sans qu'elle ne soit déverrouillée.*

Le déverrouillage consiste en une série $X_{1..N}$ d'opérations successives.

Où X est une opération faite par l'utilisateur (pression d'une touche sur le clavier, demande d'authentification, reconstitution de l'affichage, etc.).

- Le déverrouillage consiste *à demander le mot de passe utilisateur, puis à reconstituer l'écran d'affichage tel qu'il était avant le verrouillage.*

Le système doit terminer une session à la suite des $X_{1..N}$ événements.

Où X est événement provoquant la fin d'une session (temps d'inactivité pendant un verrouillage, demande de l'utilisateur, voir aussi événement d'audit, etc.).

- Le logiciel doit terminer une session *soit après un temps N de verrouillage, soit sur la demande de l'utilisateur, etc.*

1.1.2. Identification et authentification

Cette section décrit les exigences pour déterminer les moments où l'identification et l'authentification sont nécessaires. Pour qu'une authentification s'opère, il faut une identification préalable. Cela s'explique simplement comme suit: l'identification associe à un utilisateur les attributs de sécurité (nom, droit d'accès, etc.) et établit donc son identité. L'authentification contrôle l'identité de l'utilisateur, par exemple par un mécanisme de mot de passe, pour vérifier l'authenticité de celui-ci.
Les exigences sur la gestion des identifications et des authentifications doivent se trouver dans la partie 'gestion de la sécurité' à la section 'Gestion des fonctionnalités et données du système' (point 1.1.4.1.).

1.1.2.1. Identification

Concerne les exigences relatives à l'identification avant l'exécution possible d'un ensemble d'actions.

Exigences Types

Le système doit exiger que chaque utilisateur soit identifié avec succès avant d'autoriser toute action sur des fonctionnalités ou des données du système.

Dans le cas d'un système avec établissement de sessions, cette exigence décrit le moment où la première identification de l'utilisateur est requise (en générale à l'ouverture d'une session).
Dans les autres systèmes sans établissement de sessions, cette exigence n'apparaît pas, l'authentification prenant tout en charge. Par exemple, l'accès à une fonctionnalité requiert un login et un mot de passe. Le login suffit pour l'identification.

1.1.2.2. Authentification

Ensemble des exigences définissant les mécanismes d'authentification d'un utilisateur.
Les attributs de sécurité sont définis Les exigences sur la gestion des attributs de sécurité doivent se trouver dans la partie 'gestion de la sécurité' à la section 'Gestion des attributs de sécurité' (point 1.1.4.3.). L'authentification concerne l'ensemble des contrôles (attributs des droits d'accès, mot de passe, etc.) déterminant si l'utilisateur est autorisé à accéder à une ressource ou une fonctionnalité.

Exigences types

Le système doit exiger que tous les utilisateurs soient authentifiés avec succès, via des mécanismes $X_{1..N}$, avant d'autoriser toute action à des fonctionnalités ou des données du système.

Où X est mécanisme d'authentification impliquant un contrôle d'accès et (biométrie, login/password, signature électronique, etc.).

Cette exigence décrit le moment où la première authentification de l'utilisateur est requise (en générale à l'ouverture d'une session). Cette authentification est toujours précédée d'une identification.

Le système doit autoriser l'accès aux utilisateurs $X_{1..N}$ authentifiés avec succès [via des mécanismes $Y_{1..N}$,] avant d'autoriser tout accès à des fonctionnalités ou des données $Z_{1..N}$ du système.

Où X est un utilisateur du système (selon les attributs associés à l'utilisateur, etc.).
Y est un mécanisme d'authentification impliquant un contrôle d'accès (biométrie, login/password, signature électronique, etc.).
Z est une fonctionnalité ou une ressource du système.

- Le système doit autoriser l'accès aux utilisateurs $X_{1..N}$ authentifiés avec succès via des mécanismes $Y_{1..N}$, avant d'autoriser tout accès à des fonctionnalités ou des données $Z_{1..N}$ du système.

Cette exigence est relative au contrôle d'accès aux ressources et fonctionnalités durant une session ouverte.

Selon les circonstances $X_{1..N}$, le nombre de tentatives d'authentification infructueuses est limité à N essais.

Où X est une circonstance qui détermine une limite maximum du nombre de tentatives (accès à des ressources d'importance capitale, etc.).

- *Lors d'un accès à une session*, le nombre de tentatives infructueuses est limité à *N* essais.

Selon les circonstances $X_{1..N}$, le temps acceptable pour l'opération d'authentification est de maximum N secondes.

Où X est une circonstance qui détermine une limite maximum de temps d'authentification (accès à des ressources d'importance capitale, ressource non partageable, etc.).

- *Lors d'un accès au module de gestion des ventes*, le temps acceptable pour l'opération d'authentification *est de maximum 180 secondes.*

Le système doit être capable de détecter et d'empêcher l'utilisation de données d'authentification qui ont été contrefaites ou copiées.

En cas d'échec X, le système effectue une action Y.

Où X est une situation considérée comme un échec (droits d'accès non correct, etc.).
Y est une action (envoi de message, refus de l'accès, rien, etc.).

- *Après 3 tentatives d'authentification*, le logiciel *enverra un message de refus d'accès et bloquera le compte de l'utilisateur.*

1.1.3. La confidentialité du système

Cette section décrit les exigences sur les mécanismes garantissant la confidentialité des données et le respect de la vie privée.

1.1.3.1. Le cryptage

Le cryptage dépend de l'ensemble des données à chiffrer (toutes les données, seulement les données sensibles, etc.), des types algorithmes et de la longueur des clés utilisées.

Exigences types

Le cryptage de type W est utilisé pour l'ensemble des données $X_{1..N}$ [de types Y] [lors d'une action Z].

Où W est un type de cryptage (N bits, type de clé, norme, etc.).
X est une donnée à crypter.
Y est un type de données (confidentielles, non confidentielles, stratégiques, toutes les données, signature électroniques, etc.).
Z est une action (communication, transfert interne comme résultat d'une fonction, stockage, exportation, etc.).

- Le cryptage *asymétrique conforme à la norme X.509* est utilisé pour *les données qualifiées sensibles du module de gestion des ventes en ligne lors d'une communication avec le module de gestion des stocks.*

1.1.3.2. L'anonymat

Ce point décrit les exigences sur les mécanismes garantissant l'anonymat de l'utilisateur.

Exigences types

Le système doit garantir qu'un ensemble $X_{1..N}$ d'utilisateurs soit incapable de déterminer la véritable identité d'un utilisateur [lorsque celui-ci utilise une ressource ou une fonction/service].

Où X est un utilisateur du système (selon certains attributs d'utilisateur, etc.).

- Le logiciel doit garantir que *tous les simples utilisateurs* soient incapables de déterminer la véritable identité d'un autre utilisateur *lorsque celui-ci exécute une fonction du module de gestion des données personnelles.*

Le système doit fournir à l'ensemble $X_{1..N}$ d'utilisateurs un accès à un ensemble $Y_{1..N}$ de ressources ou fonctions/services sans exiger une quelconque référence à leurs véritables identités.

Où X est un utilisateur du système (selon certains attributs d'utilisateur, etc.).
Y est une ressource (donnée) du système.

- Le logiciel doit fournir *aux utilisateurs qualifiés comme simple utilisateur un accès aux fonctionnalités du module de consultation des stocks* sans exiger une quelconque référence à leurs véritables identités.

1.1.3.3. Le pseudonyme

Ce point décrit les mécanismes d'utilisation de pseudonymes pour garantir l'anonymat. Des mécanismes permettant de retrouver la véritable identité d'un utilisateur à partir d'un pseudonyme seront nécessaires si l'on souhaite établir un système de responsabilités des utilisateurs (avoir des preuves). La première exigence est relative à la possibilité d'agir sous un pseudonyme, la seconde à la réversibilité de pseudonyme et les troisième et quatrième à la possibilité d'agir sous pseudonyme via un alias.

Exigences types

Le système doit garantir qu'un ensemble $X_{1..N}$ d'utilisateurs soit incapable de déterminer le véritable nom d'un utilisateur à partir d'un pseudonyme [lorsque celui-ci utilise une ressource ou une fonction/service].

Où X est un utilisateur du système (selon certains attributs d'utilisateur, etc.).

- Le logiciel doit garantir que *tous les simples utilisateurs soient* incapables de déterminer le véritable nom d'un utilisateur à partir d'un pseudonyme *lorsque celui-ci exécute une fonction du module de gestion des données personnelles.*

Le système doit fournir à ensemble $X_{1..N}$ d'utilisateurs, la possibilité de déterminer la véritable identité d'un utilisateur à partir d'un [ensemble $Y_{1..N}$ d'] alias associé à celui-ci [et uniquement sous les conditions $Z_{1..N}$].

Où X est un utilisateur du système (selon certains attributs d'utilisateur, etc.).
Y est un alias associé à un utilisateur.
Z est un contexte d'utilisation de l'exigence (possibilité de connaître la véritable identité dans le cas d'un évènement d'audit etc.).

- Le logiciel doit fournir aux *gestionnaires du réseau,* la possibilité de déterminer la véritable identité d'un utilisateur à partir *d'un ensemble d'alias* associés à celui-ci.

Le système doit pouvoir créer un alias, accepter l'alias de l'utilisateur et pouvoir contrôler sa conformité à la métrique [(X)] utilisée pour les alias.

Où X est la métrique utilisée pour l'alias (syntaxe utilisée pour l'alias par exemple uniquement des chiffres, etc.).

- Le logiciel doit être capable de créer un alias, d'accepter l'alias de l'utilisateur et de contrôler sa conformité à la métrique *utilisée pour les alias.*

Sous les conditions $X_{1..N}$, le système doit pouvoir fournir à l'utilisateur un alias identique à un précédent. Dans les autres cas, le système fournira un alias sans relation avec les précédents alias.

Où X définit un contexte d'utilisation de l'exigence (lors d'un accès à tel fonctionnalité, consultation de données dont les droits d'accès sont limités à l'alias, etc.).

- *Lors d'un accès aux données qualifiées de sensibles et enregistrées comme propriété de l'alias,* le logiciel doit pouvoir fournir à l'utilisateur un alias identique à un précédent. Dans les autres cas, le système fournira un alias sans relation avec les précédents alias.

1.1.3.4. La non-liabilité

Ce point décrit les exigences garantissant l'incapacité d'autres utilisateurs à faire le lien entre un utilisateur et ses multiples utilisations de ressources.

Exigences types

Le système doit garantir qu'un ensemble $X_{1..N}$ d'utilisateurs ne peuvent faire le lien entre l'utilisation répétée d'ensemble $Y_{1..N}$ de fonctions/services et l'utilisateur.

Où X est un utilisateur du système (selon certains attributs d'utilisateur, etc.).
Y est une fonction du système.

- Le logiciel doit garantir *qu'un utilisateur qualifié comme simple utilisateur* ne peut faire le lien entre l'utilisation répétée *des fonctions relatives à la comptabilité analytique* et l'utilisateur.

1.1.3.5. La non-observabilité

Ce point décrit les exigences garantissant qu'un utilisateur puisse utiliser une ressource ou un service sans que d'autres, en particulier des tiers, soient capables d'observer que la ressource ou le service est en cours d'utilisation. La première exigence est relative à la non-observabilité, la seconde à l'allocation des informations ayant un impact sur la non-observabilité et la troisième à la non-observabilité sans sollicitation d'information.

Exigences types

Le système doit garantir qu'un ensemble $X_{1..N}$ d'utilisateurs ne peuvent pas observer l'utilisation d'un ensemble $Y_{1..N}$ de ressources ou de fonctions/services d'un autre ensemble $Z_{1..N}$ d'utilisateurs.

Où X est un utilisateur du système (selon certains attributs d'utilisateur, etc.).
Y est une ressource (données) du système.
Z est un utilisateur du système (selon certains attributs d'utilisateur, etc.).

- Le logiciel doit garantir que *les utilisateurs qualifiés comme simples utilisateurs* ne peuvent pas observer *l'utilisation d'aucunes ressources des autres utilisateurs qualifiés comme simples.*

Le système doit comprendre un ensemble $X_{1..N}$ de mécanismes permettant d'éviter une concentration d'un ensemble $Y_{1..N}$ d'informations relatives à la vie privée.

Où X est dispositif pour la non concentration des données (répartition des données à différents endroits, etc.).
Y est sous-ensemble de l'ensemble des informations relatives à la vie privée (données relatives au salaire, aux mails, aux documents confidentiels, etc.).

- Le logiciel doit comprendre *un système de distribution des informations* permettant d'éviter une concentration des données *personnelles des utilisateurs telles que les noms, salaires et fonctions.*

Le système doit fournir à un ensemble $X_{1..N}$ d'utilisateurs autorisés la possibilité d'observer l'utilisation d'un ensemble $Y_{1..N}$ de ressources ou de fonctions/services.

Où X est un utilisateur du système (selon certains attributs d'utilisateur, etc.).
Y est une ressource (données) du système.

- Le logiciel doit fournir *aux utilisateurs qualifiés comme administrateurs réseaux autorisés* la possibilité d'observer l'utilisation *des fonctionnalités du réseau Internet.*

1.1.4. La gestion de la sécurité

Cette section décrit les exigences garantissant une bonne gestion du système. Cette gestion passe par le paramétrage du système, par la gestion des comptes utilisateurs et par la gestion des attributs d'utilisateur.

1.1.4.1. Gestion des fonctionnalités et données du système

Ce point décrit les exigences permettant à des utilisateurs autorisés de contrôler la gestion des données et des fonctions du système. Ces fonctionnalités sont, par exemple, les fonctions d'audit ou encore d'identification et d'authentification. La première exigence est relative à la gestion du comportement des fonctionnalités et la deuxième exigence à la gestion des données du système.

Exigences types

Le système doit fournir des fonctionnalités pour permettre à un ensemble $W_{1..N}$ d'utilisateurs de pouvoir faire des modifications $X_{1..N}$ du comportement des fonctionnalités $Y_{1..N}$ [possédant un ensemble $Z_{1..N}$ de règles ou conditions de fonctionnement modifiables].

Où W est un utilisateur du système (selon certains attributs d'utilisateur, etc.).
X est fonction du système (audit, contrôle du flux d'informations, contrôle d'accès, identification et authentification, etc.).
Y est une règle associée à X (par exemple pour l'audit, qui doit-on auditer et pour quels évènements).
Z est un type de modification (déterminer le comportement, désactiver, activer ou modifier un comportement).

- Le logiciel doit fournir des fonctionnalités pour permettre *à un administrateur autorisé* de pouvoir *modifier les règles du contrôle du flux d'informations.*
- Le logiciel doit fournir des fonctionnalités pour permettre à *un administrateur autorisé* de pouvoir modifier *le degré d'audit exercé sur un utilisateur ou sur certaines fonctionnalités.*

Le système doit fournir des fonctionnalités pour permettre à un ensemble $X_{1..N}$ d'utilisateur de modifier un ensemble $Y_{1..N}$ d'attributs des fichiers du système et des utilisateurs.

Où X est un utilisateur du système (selon certains attributs d'utilisateur, etc.).
Y est un attribut d'un fichier (nom, extension, archive, caché, confidentiel, date, etc.).

- Le système doit fournir des fonctionnalités pour permettre *au propriétaire d'une donnée* de modifier *l'attribut de confidentialité de la donnée.*

1.1.4.2. Gestion des comptes utilisateurs

Ce point décrit les exigences permettant de gérer les utilisateurs et groupes d'utilisateurs du système.

Exigences types

Le système doit fournir une série $X_{1..N}$ de fonctionnalités pour la gestion des comptes utilisateurs.

Où X est une fonctionnalité pour la gestion des comptes utilisateurs (création, suppression, blocage, déblocage, historique d'accès, etc.).

Remarque : la modification des attributs de sécurité est du ressort de la partie suivante, « Gestion des attributs de sécurité ».

- Le logiciel doit fournir des fonctionnalités *pour la création, la suppression, le blocage et le déblocage des comptes, ainsi que pour l'historique d'accès etc.*, pour la gestion des comptes utilisateurs.

Le système doit fournir une série $X_{1..N}$ de fonctionnalités pour la gestion des groupes d'utilisateurs.

Où X est une fonctionnalité pour la gestion des groupes d'utilisateurs (création, modification, suppression, blocage, déblocage, historique d'accès, etc.).

- Le logiciel doit fournir des fonctionnalités *pour la création, la suppression, la modification (ajout, modification, suppression d'un membre) des groupes* pour la gestion des groupes d'utilisateurs.

1.1.4.3. Gestion des attributs de sécurité

Ensemble des exigences permettant de gérer les attributs de sécurité pour un utilisateur ou un groupe d'utilisateurs. La première exigence est relative à la gestion des attributs, et les deuxième et troisième à l'initialisation de ces attributs.

Exigences types

Le système doit pouvoir permettre à un ensemble $X_{1..N}$ d'utilisateurs d'effectuer un ensemble d'actions $Y_{1..N}$ [sur] les attributs de sécurité d'un utilisateur.

Où X est un utilisateur du système (selon certains attributs d'utilisateur, etc.).
Y est une action sur les attributs de sécurité d'utilisateur (consulter, modifier, supprimer les attributs de sécurité d'audit, du nombre de sessions, d'accès aux ressources ou données sensibles, etc.).

- Le logiciel doit pouvoir permettre *aux administrateurs système de consulter, modifier, supprimer les attributs de sécurité d'audit, du nombre de sessions* d'un utilisateur.

Lors de l'ajout d'un utilisateur [parmi un ensemble $X_{1..N}$ d'utilisateurs], chaque attribut de sécurité du nouvel utilisateur possède une valeur par défaut [propre à l'ensemble $X_{1..N}$].

Où X est un utilisateur du système (selon certains attributs d'utilisateur, etc.).

- Lors de l'ajout d'un utilisateur *qualifié d'utilisateur simple*, chaque attribut de sécurité du nouvel utilisateur possède une valeur par défaut.

Le système doit permettre à un ensemble $X_{1..N}$ d'utilisateurs d'effectuer un ensemble d'actions $Y_{1..N}$ sur les valeurs par défaut des attributs de sécurité [tel que $Z_{1..N}$] .

Où X est un utilisateur du système (selon certains attributs d'utilisateur, etc.).
Y est une action sur les attributs de sécurité d'utilisateur (consulter, modifier, réinitialiser à la valeur par défaut, etc.).
Z est un attribut de sécurité d'un utilisateur (droits d'accès, évènements à auditer, etc.).

- Le logiciel doit permettre *aux administrateurs système de consulter, modifier* la valeur par défaut des attributs *de sécurité d'audit, du nombre de sessions, d'accès aux ressources ou données sensibles, etc.*

Le système doit contenir un ensemble $X_{1..N}$ de valeurs possibles [(restrictives, permissives, ou autres propriétés)] pour les attributs de sécurité, et doit garantir qu'aucun utilisateur ne puisse y déroger.

Où X est une valeur possible d'un attribut de sécurité (par exemple, le mot de passe d'un utilisateur doit être un entier de quatre chiffres, etc.).

- Le logiciel doit contenir un *ensemble de valeurs possibles (par exemple, la syntaxe d'un mot de passe est composé de N chiffres)* pour les attributs de sécurité, et doit garantir qu'aucun utilisateur ne puisse y déroger.

1.1.5. L'intégrité des fichiers

Cette section décrit les exigences relatives à l'intégrité des bases de données et autres fichiers du système. Que doit-on faire pour éviter que le système ne corrompe ou ne perde les données stockées en cas d'événements inattendus (comme par exemple une attaque de l'extérieur ou encore une erreur non intentionnelle de l'intérieur).

1.1.5.1. Les back-up, imports et exports

Ce point présente les exigences relatives aux sauvegardes des données et aux transferts de celles-ci sur d'autres supports ou vers d'autres logiciels. La première exigence est relative aux sauvegardes, la seconde à l'exportation de données et la troisième à l'importation de données.

Exigences types

[Tous les W période de temps,] le système doit sauvegarder sur un support de stockage de type X, les informations [appartenant à l'ensemble $Y_{1..N}$ d'utilisateurs ou de fonctionnalités] relatives aux données de type Z [sans devoir arrêter l'exploitation des autres applications].

Où W est une unité de temps (toutes les 24 heures, etc.).
X est un type de support de stockage (disque optique, disque magnétique, bande magnétique, etc.).
Y est un utilisateur du système (selon certains attributs d'utilisateur, etc.).
Z est un type de données (avec un certain attribut, etc.).

- *Toutes les 24 heures*, le système doit sauvegarder sur un support *optique*, les informations *des administrateurs* relatives aux *données confidentielles*.

Le système doit appliquer les contrôles d'accès et de flux d'informations lors de l'exportation de données d'un utilisateur contrôlé par le système, vers l'extérieur du système.

Le système doit exporter les données de l'utilisateur avec [(ou) sans] les attributs de sécurité qui leur sont associés.

1.1.5.2. L'annulation

Ce point décrit les exigences fournissant à l'utilisateur la capacité d'annuler les effets d'une opération effectuée, ou une série d'opérations effectuées, dans une limite déterminée (limite de temps ou d'un état des actions effectuées précédemment connue) afin de préserver l'intégrité des données de l'utilisateur.

Exigences types

Le système doit fournir un mécanisme X d'annulation pour l'ensemble des fonctionnalités suivantes : $Y_{1..N}$.

Où X est un dispositif d'annulation (annulation des N dernières opérations, point de sauvegarde pour restauration).
Y est une fonctionnalité du système.

- Le logiciel doit fournir un système de points de sauvegarde *pour toutes les fonctionnalités du système.*
- Le logiciel doit fournir des opérations d'annulation d'opérations *pour la fonctionnalité d'introduction des données dans le module de la gestion des stocks.*

Le système doit autoriser l'annulation selon l'ensemble $X_{1..N}$ constituant les limites de l'annulation du mécanisme Y.

Où X est une limite de l'annulation (uniquement les N dernières opérations, nombre de points de restauration autorisés, etc.).
Y est un dispositif d'annulation (annulation des N dernières opérations, point de sauvegarde pour restauration).

- Le logiciel doit autoriser l'annulation *uniquement de la dernière opération.*
- Le logiciel doit autoriser l'annulation *par un mécanisme de restauration d'un point de sauvegarde.*

1.1.5.3. Politique et fonctions du contrôle du flux d'informations entrant et sortant.

Ce point décrit les exigences pour le filtrage de l'information garantissant l'intégrité des données lors d'un flux d'information vers ou en provenance d'un utilisateur.

Exigences types

Le système doit appliquer un ensemble $X_{1..N}$ de règles pour le contrôle du flux d'informations à l'ensemble $Y_{1..N}$ d'utilisateurs.

Où X est une règle pour le contrôle du flux d'informations (filtrage des entrées et sorties de certains fichiers, filtrage des spams, autres règles pour les virus et chevaux de Troie, etc.).
Y est soit un utilisateur, soit une fonctionnalité du système qui déclenche un transfert d'informations contrôlées vers ou en provenance d'utilisateurs contrôlés par le système.

- Le logiciel doit appliquer *un filtrage des sorties des fichiers confidentiels* pour le contrôle du flux d'informations *à l'ensemble des utilisateurs.*

Dans les conditions $X_{1..N}$, le système doit autoriser le flux d'informations [pour l'ensemble $Y_{1..N}$ d'utilisateurs] selon les règles suivantes : $Z_{1..N}$.

Où X est une condition précisant un contexte pour autoriser le flux d'informations (communication par canaux sécurisés, etc.).
Y est un utilisateur du système (selon certains attributs d'utilisateur, etc.).
Z est une règle pour le contrôle du flux d'informations (comparaison des attributs de sécurité des utilisateurs et des données, filtrage des entrées et sorties de certains fichiers, filtrage des spams, autres règles pour les virus et chevaux de Troie, etc.).

- *Lors de communication sécurisée avec une entité extérieure au système,* le logiciel doit autoriser le flux d'informations *de tous les utilisateurs* selon les règles suivantes *: un filtrage des informations sera effectué pour éviter que des données confidentielles ne soient échangées sans les droits requis.*

Le système doit interdire le flux d'informations lorsqu'une règle X est outrepassée [pour l'ensemble $Y_{1..N}$ d'utilisateurs].

Où X est une règle pour le contrôle du flux d'informations (comparer les attributs de sécurité des utilisateurs et des données, filtrage des entrées et sorties de certains fichiers, filtrage des spams, autres règles pour les virus et chevaux de Troie, etc.).
Y est un utilisateur du système (selon certains attributs d'utilisateur, etc.).

- Le logiciel doit interdire le flux d'informations lorsque *le contrôle du flux détecte un risque potentiel de contamination par un virus.*

1.1.5.4. Contrôle d'intégrité des données stockées

Ce point décrit les exigences permettant de garantir que les données stockées dans le système (mémoire ou support de stockage) sont intègres. La première exigence est relative au contrôle de l'intégrité et la seconde aux actions à entreprendre.

Exigences types

Le système exécute des contrôles d'intégrité de type $X_{1..N}$ pour toutes les données ayant les attributs $Y_{1..N}$.

Où X est vérificateur d'intégrité (Checksum, CRC,...).
Y est un attribut de données (nom, date, caché, version, confidentiel, archive, nom de la fonctionnalité ou utilisateur auquel elles se rapportent, etc.).

- Le logiciel doit exécuter des contrôles d'intégrité sur les CRC pour toutes les données ayant *les attributs confidentialité.*

Le système doit effectuer un ensemble $X_{1..N}$ d'actions lorsque celui-ci détecte une erreur d'intégrité.

Où X est une action du système lors d'une détection d'erreur (avertir l'utilisateur, rendre l'accès impossible, supprimer la donnée, réparer la donnée, etc.).

- Le logiciel doit soit *avertir l'utilisateur,* soit *s'il en est capable, réparer la donnée, sinon la supprimer avec approbation de l'utilisateur* lorsque celui-ci détecte une erreur d'intégrité.

1.1.5.5. Protection des informations résiduelles

Ce point décrit les exigences garantissant que les données supprimées ou effacées sont inaccessibles.

Exigence type

Le système doit garantir que l'ensemble des données $X_{1..N}$ sera définitivement inaccessible après effacement.

Où X est une donnée du système (donnée confidentielle, donnée d'un utilisateur, etc.).

- Le logiciel doit garantir que *l'ensemble des données qualifiées comme sensibles* sera définitivement inaccessible après effacement.

1.1.6. Non répudiation des échanges et des transactions

Cette section concerne les exigences permettant de fournir des preuves des actions et des transactions effectuées. Ces exigences garantissent que le récepteur ne peut contester la réception du message, tout comme l'émetteur ne peut contester l'émission du message.

1.1.6.1. Emetteur

Exigence type

Sur demande de personnes $X_{1..N}$, le système doit pouvoir fournir la preuve de la réception pour l'ensemble $Y_{1..N}$ des informations transmises.

Où X est un utilisateur du système (selon certains attributs d'utilisateur, etc.).
Y est une information reçue (selon la date, le nom des informations, l'émetteur des informations, etc.).

- Sur demande *de l'envoyeur ou de tout autre personne autorisée*, le système doit pouvoir fournir la preuve de la réception pour des *informations confidentielles* transmises.

1.1.6.2. Récepteur

Exigence type

Sur la demande de personnes $X_{1..N}$, le système doit pouvoir fournir la preuve de l'origine pour l'ensemble $Y_{1..N}$ des informations reçues.

Où X est un utilisateur du système (selon certains attributs d'utilisateur, etc.).
Y est une information reçue (selon la date, le nom des informations, l'émetteur des informations, etc.).

- Sur demande *du récepteur ou de tout autre personne autorisée*, le système doit pouvoir fournir la preuve de l'origine pour des *informations confidentielles* reçues.

1.1.7. L'audit

L'audit concerne toutes les activités d'identification, d'enregistrement, de stockage et d'analyse de l'information liées à la sécurité. Les enregistrements d'audit peuvent être examinés par la suite en vue de détecter les activités relatives à la sécurité et aux personnes qui en sont responsables.

1.1.7.1. La réponse automatique

Ce point décrit les exigences définissant l'ensemble des actions automatiques à entreprendre en cas de détection de violation potentielle du système.

Exigence type

Le système doit effectuer un ensemble $X_{1..N}$ d'actions lors d'une détection de violation potentielle de la sécurité.

Où X est une action (terminer la session, bloquer la session, bloquer les communications, etc.).

- Le système doit *terminer la session de l'utilisateur responsable* lors d'une détection de violation potentielle de la sécurité.

1.1.7.2. La génération de données

Ce point concerne les exigences décrivant les conditions de génération des données lors de la détection d'événements contrôlés par le système. Ces exigences décrivent le type d'événement qui doit être audité et l'ensemble minimal des informations liées à l'audit qui devrait se retrouver dans les enregistrements d'audit. Dans cette partie, il y a d'un côté, la génération de données (première et deuxième exigence) et de l'autre, la liaison de celles-ci avec l'identification de l'utilisateur.

Exigences types

Le système doit pouvoir générer un enregistrement d'audit pour les évènements à auditer : **- De démarrage et d'arrêt des fonctions d'audit,** **- Pour l'ensemble $X_{1..N}$ des évènements à auditer ayant un niveau d'audit Y [ou pour tout autre ensemble $Z_{1..N}$ d'évènements à auditer spécifiquement définis].**

Où X est un évènement à auditer avec un niveau Y (accès à une ressource, transfert de fichiers, communication avec l'extérieur du système, etc.).
Y est un niveau d'audit (minimum, élémentaire, détaillé, non spécifié, ou autre métrique).
Z est un évènement à auditer spécifiquement défini (par exemple n'ayant pas le niveau Y mais considéré comme important à enregistrer).

- Le système doit pouvoir générer un enregistrement d'audit :

- pour les évènements *à auditer* de démarrage et d'arrêt des fonctions d'audit,
- *pour tous les évènements à auditer* avec un niveau d'audit *détaillé.*

Dans chaque enregistrement d'audit, le système doit enregistrer au minimum :
- la date et heure de l'évènement,
- l'identité de l'utilisateur,
- le résultat [(succès ou échec)] de l'évènement, le type X d'évènement d'audit,
- pour chaque type X d'évènement d'audit [(sur base des évènements à auditer contenus dans les composants fonctionnels du système)], l'ensemble $Y_{1..N}$ des informations d'audit pertinentes.

Où X est un type d'évènement (accès refusé lors d'une authentification, attaque potentielle ou réelle, intrusion potentielle ou réelle, transfert non autorisé de fichiers confidentiels, échec d'un accès au lecteur réseau, etc.).
Y est une information pertinente et spécifique à la fonctionnalité (lors d'une attaque externe, on enregistre, si possible, les informations de la source ainsi que le type d'attaque, etc.).

Le système doit pouvoir relier chaque évènement à auditer avec l'utilisateur responsable de cet évènement.

1.1.7.3. L'analyse

Ensemble des exigences des moyens mis en œuvre pour l'analyse des activités du système et des informations de l'audit. L'analyse consiste à rechercher les possibles ou réelles menaces contre le système. Cette analyse peut déboucher sur la détection d'intrusions et sur une série d'actions automatiques en cas de violation imminente de la sécurité. Les deux premières exigences sont relatives à l'analyse de violations potentielles. Les trois suivantes à la détection d'anomalies basées sur un profil (qui est un historique des comportements des utilisateurs). En effet, il s'agit aussi de détecter les anomalies des comportements des utilisateurs cibles. Les deux dernières exigences concernent l'heuristique des attaques (simples et complexes).

Exigences types

Le système doit pouvoir appliquer un ensemble de règles en surveillant les évènements à auditer et indiquer, en fonction de celles-ci, une violation potentielle du système.

Les règles à appliquer pour la surveillance sont l'accumulation ou la combinaison/succession d'un ensemble $X_{1..N}$ d'évènements à auditer considérés comme des violations potentielles de la sécurité.

Où X est un évènement à auditer (accès à une ressource non autorisée, transfert de fichiers, communication avec l'extérieur du système, etc.).

- Les règles à appliquer pour la surveillance sont, pour l'accumulation d'évènements, *une succession de tentatives infructueuses puis fructueuses de login à partir d'Internet, suivi d'une consultation de données sensibles, etc.* considérés comme des violations potentielles de la sécurité.

Le système doit pouvoir maintenir des profils d'utilisation pour chaque utilisateur. Ces profils d'utilisations individuels représentent un historique des comportements d'un ensemble $X_{1..N}$ d'utilisateurs cibles.

Où X est un utilisateur du système (selon certains attributs d'utilisateur, etc.).

- Le système doit pouvoir maintenir des profils d'utilisation pour chaque utilisateur. Ces profils individuels représentent un historique des comportements *des utilisateurs ayant accès au module de gestion des ventes.*

Le système doit pouvoir maintenir un indice de représentativité pour chaque utilisateur dont l'activité est enregistrée dans un profil. Cet indice indique le degré avec lequel l'activité actuelle de l'utilisateur diffère des modèles d'utilisation représentés dans le profil.

Le système doit être capable d'indiquer une violation imminente lorsque l'indice de représentativité dépasse le seuil limite d'une des conditions de l'ensemble $X_{1..N}$ des conditions d'activités anormales.

Où X est un seuil en dessous duquel une activité est normale (succession d'événements inhabituels, nombre de tentatives d'accès aboutissant à un échec, etc.).

- Le système doit être capable d'indiquer une violation imminente lorsque l'indice de représentativité dépasse le seuil limite *du nombre N d'accès aux ressources confidentielles non consultées habituellement.*

Le système doit pouvoir maintenir une représentation en interne de l'ensemble des évènements à auditer et de l'ensemble des enchaînements d'évènements qui peuvent indiquer une violation du système.

Le système doit pouvoir indiquer une violation imminente quand l'activité du système correspond à un évènement ou enchaînement d'évènements caractéristiques indiquant une violation potentielle du système.

1.1.7.4. La revue d'audit

Ce point décrit les exigences concernant les outils d'audit mis à la disposition des utilisateurs autorisés pour aider à l'examen des données d'audit. Les deux premières exigences sont relatives à la revue de l'audit et la troisième à la sélection.

Exigences types

Le système doit permettre à un ensemble $X_{1..N}$ d'utilisateurs autorisés de lire l'ensemble $Y_{1..N}$ des informations dans les enregistrements d'audit.

Où X est un utilisateur du système (selon certains attributs d'utilisateur, etc.).
Y est une information d'audit (date et heure de l'évènement, l'identité de l'utilisateur, le résultat (succès ou échec) de l'évènement, le type d'évènement d'audit, et autres informations).

- Le système doit permettre *aux administrateurs autorisés* de lire *toutes les informations* des enregistrements d'audit.

Le système doit présenter les informations d'audit d'une façon telle qu'elle permet à l'utilisateur de pouvoir les interpréter.

Le système doit pouvoir effectuer un ensemble $X_{1..N}$ d'opérations sur les données de l'audit selon un ensemble $Y_{1..N}$ de critères liés logiquement à ces opérations.

Où X est une opération (de recherche, de tri ou d'ordonnancement).
Y est un critère lié à l'opération X (recherche selon le nombre d'occurrences d'un évènement, tri selon la date, etc.).

- Le système doit pouvoir permettre d'effectuer *un tri* sur les données de l'audit selon le *nombre d'occurrences d'un même évènement.*

1.1.7.5. La possibilité de sélection des évènements d'audit

Ce point décrit les exigences qui déterminent les évènements à auditer lorsque le système fonctionne.

Exigence type

Le système doit pouvoir inclure ou exclure des évènements $X_{1..N}$ dans l'ensemble des évènements à auditer en fonction d'un ensemble $Y_{1..N}$ d'attributs.

Où X est un évènement à auditer (accès non autorisé à une ressource, détection d'évènements anormaux lors de l'exploitation, etc.).
Y est un attribut pour l'inclusion ou l'exclusion d'un évènement (identité de la ressource, identité de l'utilisateur, identité de l'hôte, type d'évènement ou autres).

- Le système doit pouvoir inclure ou exclure des évènements de l'ensemble des évènements à auditer en fonction *de l'identité de l'utilisateur*.

1.1.7.6. L'enregistrement d'évènements d'audit de sécurité

Ce point décrit les exigences pour que le système soit capable de créer et de maintenir une trace d'audit sûre. Les deux premières exigences concernent la protection du stockage, la troisième est relative à la disponibilité des données d'audit, la quatrième à la perte de données d'audit et la cinquième à la prévention de perte de données d'audit.

Exigences types

Le système doit protéger les enregistrements d'audit contre toute suppression non autorisée.

Le système doit pouvoir empêcher [(ou détecter)] des modifications effectuées sur les enregistrements d'audit.

- Le système doit pouvoir *détecter* des modifications effectuées sur les enregistrements d'audit.

Selon l'ensemble $X_{1..N}$ des conditions, le système doit garantir que la métrique Y des enregistrements d'audit sera maintenue.

Où X est une condition déterminant un contexte pouvant aboutir à une défaillance des enregistrements (soit dépassement de la capacité du stockage, défaillance, attaque).
Y est un métrique pour les enregistrements (la syntaxe utilisée dans une trace d'enregistrement, par exemple, quelles informations et l'ordre de celles-ci, etc.).

- *En cas de dépassement de la capacité du stockage*, le système doit garantir que la métrique des traces d'enregistrements d'audit *(qui est structurée dans cet ordre : la date, le type d'événement, l'identification du responsable, et puis les informations spécifiques au contexte d'utilisation)* sera maintenue.

Le système doit pouvoir effectuer une série $X_{1..N}$ d'actions dans le cas d'une défaillance possible dans le stockage des données d'audit, si le seuil critique de trace d'audit dépasse la limite Y.

Où X est une action (backup, message d'alerte, etc.).
Y est une limite de taille (octets, mégas, ou autre).

- Le système devra effectuer *un backup des traces sur un support de secours* dans le cas d'une défaillance possible dans le stockage des données d'audit, si la trace d'audit dépasse la limite *de 350 mégas*.

Quand la trace de l'audit est pleine, le système doit entreprendre une ou plusieurs actions $X_{1..N}$.

Où X est une action que le système prend lorsque la trace est pleine (ignorer les évènements à auditer, empêcher les événements à auditer autres que ceux provoqués par des utilisateurs ayant des autorisations spécifiques, écraser les enregistrements et/ou autre action spécifique).

- Quand la trace de l'audit est pleine, le système doit *ignorer les évènements à auditer*.

1.2. Infrastructure et exigences techniques

1.2.1. Systèmes d'exploitation

Cette section décrit les exigences relatives aux systèmes d'exploitation utilisés dans l'entreprise.

Exigence type

Le système doit pouvoir tourner sur l'ensemble $X_{1..N}$ de système d'exploitation.

Où X est un système d'exploitation (Linux, Windows 2000 , Windows NT, etc..).

1.2.2. Réutilisation des équipements informatiques existants (excepté protocoles réseaux)

1.2.2.1. Réutilisation Hardware

Ce point décrit les exigences précisant le degré de réutilisation du matériel hardware actuel. Pour chaque équipement, il est conseillé une description suffisamment détaillée pour permettre au fournisseur de disposer d'un maximum d'informations et de déterminer la compatibilité de son logiciel avec le matériel à réutiliser.

Exigence type

Le système doit pouvoir réutiliser l'ensemble $X_{1..N}$ des équipements hardware actuels.

Où X est un composant de l'ensemble des composants utilisés par l'entreprise (serveur, poste client, infrastructure réseau, base de données, etc.).

- Le logiciel doit pouvoir réutiliser *les postes clients (Pentium III 800, 256 Mo RAM,etc.) et serveur.*

A préciser :

- Si l'application doit s'interfacer avec un système embarqué, d'autres contraintes peuvent s'ajouter (temps de réponse, consommation processeur et batterie, etc.).

Conseils pour la description de composants d'infrastructure

Il ne s'agit pas ici d'exigences mais bien d'informations à indiquer pour décrire le matériel hardware :

- Pour les postes clients : type, CPU, ram, capacité HDD, SE, applications, etc.
- Pour les serveurs : type, CPU, ram, capacité HDD, système d'exploitation utilisé, anti-virus, back-up (support, périodicité), son utilisation), etc.
- Pour les réseaux : le nombre de hub, de switch et routeur, les types de câbles, les types de réseaux, les lignes de communication, etc.
- Pour les bases de données : leurs capacités, leurs performances, leurs fréquences d'utilisation, les conditions de conservation de données, etc.

1.2.3. Réutilisation des réseaux

Cette section décrit les exigences relatives aux protocoles réseaux et aux types de réseaux.

Exigence type

Le système doit pouvoir fonctionner sur l'ensemble $X_{1..N}$ de type de réseaux et sur l'ensemble $Y_{1..N}$ des protocoles réseaux utilisés par l'entreprise.

Où X est un type de réseaux utilisés par l'entreprise (réseau filaire comme Ethernet, réseau sans fil comme WiFi, etc.).
Y est un protocole utilisé par l'entreprise[5] (TCP IP, UDP, SMTP, FTP, etc.).

- Le logiciel doit pouvoir fonctionner *sur le réseau Client/Serveur* avec comme protocoles *TCP IP, SMTP.*

1.2.4. Procédure d'installation et de test

Cette section concerne les exigences décrivant les contraintes d'installation du logiciel, et les jeux de test à fournir.

1.2.4.1. Contraintes d'installation

Ces contraintes sont déterminées en fonction du logiciel à acquérir. Elles concernent, d'une part, les contraintes d'installation du logiciel sur un poste client et un poste serveur, et d'autre part, les contraintes de configuration du logiciel d'un poste client et d'un poste serveur.

[5] Source http://www.fing.org/index.php?num=3869,3,164,5

Exigences types

L'installation du système doit être à la charge du fournisseur pour un nombre N de postes clients.

Où N est un nombre entier.

La configuration du système doit être à la charge du fournisseur pour un nombre N de postes clients.

Où N est un nombre entier.

L'installation du système doit être à la charge du fournisseur pour un nombre N de serveurs.

Où N est un nombre entier.

La configuration du système doit être à la charge du fournisseur pour un nombre N de serveurs.

Où N est un nombre entier.

Le fournisseur doit spécifier les procédures d'installation et de configuration du système pour les postes clients et serveurs.

1.2.4.2. Test

Ce point décrit les exigences relatives aux jeux de tests software et hardware.

Exigence type

Le fournisseur doit exécuter un ensemble $X_{1..N}$ de jeux de tests de degré Y pour l'ensemble $Z_{1..N}$ des fonctionnalités.

Où X est un jeu de tests (sur les valeurs ou sur les performances lors de conditions d'utilisation critique de la fonctionnalité X).
Y est un degré déterminant la qualité des tests (nombreux tests pertinents, etc.).
Z est une fonctionnalité du système.

- Le fournisseur doit exécuter *un ensemble de jeux de tests très complets pour les données critiques de la gestion des informations pertinentes des clients.*

1.3. Performances du système

La section performances doit répertorier toutes les contraintes liées aux performances du logiciel/système désiré. Ces performances sont souvent essentielles au bon fonctionnement du logiciel/système. Les exigences de vitesses sont par exemple très importantes pour une application temps réel.

1.3.1. La vitesse

Cette section décrit les exigences sur les contraintes de temps de réponse.

1.3.1.1. Les interfaces

Ce point décrit les exigences relatives au temps pour un passage d'une interface à une autre.

Exigence type

Le système doit pouvoir fournir pour l'ensemble $X_{1..N}$ d'interfaces un temps de passage d'un écran à un autre inférieur à Y.

Où X est une interface du système.
Y est une unité de temps (minutes, secondes, etc.).

- Le système doit pouvoir fournir pour *toutes les fenêtres d'IHM* un temps de passage d'un écran à un autre inférieur à *N secondes*.

1.3.1.2. Temps de réponse

L'exigence de cette section concerne un temps garanti pour obtenir les résultats d'une fonctionnalité. Ce temps inclut donc tous les accès nécessaires (accès base de données, accès aux fonctionnalités, et autre accès hardware comme le réseau, etc.)

Exigence type

Le système doit garantir un temps de réponses inférieur à X pour obtenir les résultats d'une fonctionnalité Y [ou d'un ensemble de fonctionnalité].

Où X est un seuil maximum de temps de réponses (minutes, secondes, etc.).
Y est une fonctionnalité (recherche de données, etc.).

- Le logiciel doit garantir un temps de réponse inférieur à *N secondes* pour obtenir les résultats de la *fonctionnalité évaluant les stocks et approvisionnements nécessaires en fonction des futures ventes estimées.*
- Le logiciel doit garantir un temps de réponse inférieur à *N secondes* pour l'ensemble des fonctionnalités de la gestion de production.

1.3.2. La précision

Les exigences de précision ont pour but de quantifier la précision désirée des résultats afin d'éviter toute ambiguïté ou erreur.

Exigence type

X doit être précis à Y près.

Où X représente une donnée ou un résultat d'une fonctionnalité.
Y représente un degré de précision ou une marge d'erreur.

- Les sommes monétaires doivent être précises à 2 décimales.
- Les salaires seront calculés à 1 Euro près.

1.3.3. Capacité de traitement et stockage de données

Cette section décrit les exigences visant à s'assurer que le logiciel sera effectivement capable de faire face à la charge de travail qui lui sera demandée. Les variations de cette charge dans le temps seront explicitées dans la section suivante.

2.3.3.1 *Nombre d'utilisateurs*

Exigence type

Le système supportera N utilisateurs simultanément [lorsque la période de temps est X].

Où N est un nombre entier.

X est une période de temps (entre N heures et M heures, à telle date, lors du lancement et de la fermeture, suite à tel évènement, etc.).

- Le système supportera *500 utilisateurs* simultanément.

2.3.3.2 *Nombre de tâches*

Exigence type

Le système est capable de traiter $X_{1..N}$ tâches simultanées [lorsque la période de temps est Y].

Où X est un ensemble de tâches (différentes ou identiques) à effectuer simultanément.
Y est une période de temps (entre N heures et M heures, à telle date, lors du lancement et de la fermeture, suite à tel évènement, etc.).

- Le système est capable de traiter *12 demandes* simultanément *entre 8 heures et 16 heures.*

2.3.3.3 *Volume des données*

Exigence type

Le système doit être capable de traiter un volume X de données [lorsque la période de temps est Y].

Où X est un volume d'informations.
Y est une période de temps (entre N heures et M heures, à telle date, lors du lancement et de la fermeture, suite à tel évènement, etc.).

- Le système doit être capable de *traiter 600 clients.*

1.3.4. Adaptation à une montée en charge

Ce point vise à vérifier que le système sera capable de faire face à une augmentation des besoins. Les exigences types de cette section sont donc identiques à celles de la section précédente mais suivies d'une date ou d'une période de temps future.

- Le logiciel supportera 300 utilisateurs simultanés en 2005.
- Le logiciel sera capable de traiter 50 transactions par seconde entre 16h et 18h dans deux mois.
- Le volume des données augmentera dans une certaine période.

1.4. Disponibilité

Concerne les exigences relatives à la disponibilité des ressources et des services d'un système.
A part les exigences de ce point, d'autres mesures contribuent à assurer la disponibilité : se protéger contre les virus, se doter de systèmes performants capables de gérer un trafic intense sur le réseau et posséder des serveurs redondants.

1.4.1. Tolérance aux pannes

Cette section décrit les exigences déterminant le degré désiré de tolérance aux fautes du système. La première exigence est relative à la tolérance limite, la seconde à la tolérance aux fautes limitée à certaines défaillances, la troisième aux périodes de disponibilité et la quatrième à la disponibilité pendant les tests en plate-forme.

Exigences types

Le système doit garantir la disponibilité d'un ensemble $X_{1..N}$ de capacité lorsqu'un ou plusieurs éléments de l'ensemble $Y_{1..N}$ des défaillances surviennent.

Où X est une capacité du système (accès à une ressource stratégique, accès au réseau, etc.).
Y est une défaillance (lors de l'exécution de certaines fonctionnalités non stratégiques, lors de l'accès à des ressources non essentielles du système, etc.).

- Le système doit garantir la disponibilité *des capacités de transfert et sauvegarde des données en cours d'utilisation* soit lorsqu'une erreur *lors de l'exécution d'une fonctionnalité du module de gestion des données personnelles* survient, soit *lorsqu'une erreur suite à une surcharge du système* survient.

L'ensemble $X_{1..N}$ des fonctionnalités du système doit être disponible pendant une période temps Y.

Où X est une fonctionnalité du système (par exemple, l'accès aux informations pertinentes des clients, le planning des ordres de fabrication, la comptabilité analytique, etc.).
Y est une période de temps d'activités du système (24/24h, ou encore 8 heures par jour, jours d'inactivité, période de fêtes, etc.).

- Les fonctionnalités critiques du système doivent être disponibles *entre 8 heures et 19 heures sauf le week-end.*

1.4.2. Priorité de service

Cette section est relative aux priorités à accorder aux fonctionnalités pour l'accès aux ressources. Les ressources reprises dans cette partie sont appelées ressources contrôlées. Il s'agit des ressources hardware (permettant d'exécuter une tâche, par exemple les processeurs, réseaux, etc.).

Exigences types

Le système doit attribuer une priorité X à un ensemble $Y_{1..N}$ de fonctionnalités du système.

Où X est le degré de priorité accordé (par exemple, faible-moyen-forte, ou autres systèmes de mesure).
Y est une fonctionnalités du système.

- Le système doit attribuer une priorité *forte* à la fonctionnalité de *gestion des comptes clients*.

Le système doit garantir que chaque accès à un ensemble $X_{1..N}$ de ressources contrôlées doit être accordé sur base des priorités à la fonctionnalité essayant d'y avoir accès.

Où X est une ressource du système dont la priorité est contrôlée (concerne toutes les ressources permettant d'exécuter une tâche, par exemple les processeurs, réseaux, etc.).

- Le logiciel doit garantir que chaque accès *à toutes ressources partageables* doit être accordé sur base des priorités conférées à la fonctionnalité essayant d'y avoir accès.

1.4.3. Allocation des ressources

Cette section décrit les exigences permettant de contrôler la répartition d'allocation des ressources, rendant impossible tous dénis de service suite à la monopolisation non autorisée d'une ressource.

Exigence type

Le système doit limiter l'utilisation de l'ensemble de ressources $Y_{1..N}$ au quota X afin de s'assurer que la fonctionnalité Z ne les monopolise pas.

Où X est un quota (temps inférieur/supérieur à N, etc.).
Y est une ressource du système (concerne toutes les ressources permettant d'exécuter une tâche, par exemple les processeurs, réseaux, etc.).
Z est une fonctionnalité du système.

- Le système doit limiter l'utilisation du *serveur principal* à *maximum 10 secondes* afin de s'assurer que les *sauvegardes secondaires* ne le monopolisent pas.

1.5. Fiabilité

Aptitude du logiciel à maintenir son niveau de performance sous des conditions spécifiées pour une durée spécifiée.

1.5.1. Temps moyen entre deux pannes

Exigence type

L'ensemble $X_{1..N}$ des fonctionnalités du système doit avoir un MTBF [(*Mean Time Between Failure*)] de maximum Y.

Où X est une période de temps entre les pannes (mois, année, etc.).
Y est une fonctionnalité du système (par exemple, la gestion des ventes, la gestion de production, la gestion des stocks et des approvisionnements, etc.).

- La fonctionnalité de *gestion de la production* doit avoir un temps moyen de réparation de *30 minutes* car cette fonctionnalité est essentielle à l'activité de notre entreprise.

1.5.2. Temps d'action pour la réparation des défaillances

Cette section définit les temps maximaux pour la réalisation du dépannage.

Exigence type

La réparation d'un ensemble $X_{1..N}$ de défaillance doit avoir un MTTR [(*Mean Time To Repair*)] de maximum Y.

Où X est une défaillance (d'une fonctionnalité, d'un disque dur, etc.).
Y est une période de temps moyen pour la réparation des dégâts (heures, jours, etc.).

- Le remplacement d'un serveur devra se faire endéans les 24 heures.
- Le remplacement d'un disque dur doit se faire endéans les 12 heures.

1.5.3. Journal des problèmes

Cette section décrit les exigences servant à assurer un certain niveau de traçabilité d'évènements anormaux. Ces exigences détermineront la sélection des événements d'audit pour les événements liés à la fiabilité.

Exigence type

Une traçabilité sera mise en place pour un ensemble $X_{1..N}$ d'évènements anormaux.

Où X est un évènement d'audit lié à des erreurs de programmation ou de traitement.

- Une traçabilité sera mise en place *pour enregistrer les erreurs de programmation donnant lieu à des quantités négatives dans les chiffres de production.*

1.6. Maintenance

1.6.1. Facilité de maintenance du produit

Cette section décrit les exigences permettant de faciliter la maintenance du produit. Il convient donc dans cette partie de fournir une estimation du temps nécessaire pour faire les modifications nécessaires sur le produit.

1.6.1.1. Personnes chargées de maintenance

Ce point décrit les exigences spécifiant pour chaque partie du système les utilisateurs ou autres personnes responsables de la maintenance.

Exigence type

L'ensemble $X_{1..N}$ d'objets de maintenance devra être maintenu par $Y_{1..N}$.

Où X est un objet de maintenance (base de données, maintenance logiciel, etc.).
Y est une personne pouvant être chargée de la maintenance du système (personne qui participé à la création du produit, qui a commandé le produit, utilisateur final, développeur interne, utilisateur ayant suivi une formation spécifique pour la maintenance du produit, etc.).

- L'ensemble du système devra être maintenu par les développeurs de notre entreprise.

- La maintenance des bases de données devra être maintenue par l'entreprise.

1.6.1.2. Amélioration de la facilité de maintenance

Exigences types

Le système sera divisé en ensemble de $X_{1..N}$ modules [et la maintenance d'un module peut s'exécuter indépendamment des autres modules].

Où X est un module (production, gestion, etc.).

- Le système sera divisé en *un module administrateur et un module utilisateur.* La maintenance d'un module pourra s'exécuter indépendamment des autres modules.

L'ensemble des documentations $X_{1..N}$ pour la maintenance doit être fourni et remis à jour.

Où X est un support de documentation (spécification du logiciel, ou d'une partie de celui-ci, etc.).

1.6.2. Planning des fréquences et garantie de mise à jour

Cette section définit les exigences relatives au planning des nouvelles versions (fréquence des updates) du produit et la forme de ces nouvelles versions.

Exigences types

Une mise à jour sera disponible tous les X sous la forme Y.

Où X est une unité de temps.
Y est la forme des mises à jour (patch auto-exécutable accessible sur le Web, etc.).

- Une mise à jour sera disponible *tous les ans sous la forme de patch auto-exécutable fournie par le vendeur.*

Le fournisseur de logiciel doit fournir un ensemble de supports $X_{1..N}$ pour l'installation des mises à jour.

Où X est un ensemble de supports et de garanties pour les mises à jour (supports écrits comme manuel, site Internet, FAQ, GMAO etc. et garanties temporelles, garantie de qualité des mises à jour, garantie de l'intégrité des données après mise à jour, garantie de disponibilité, etc.).

- Le logiciel doit fournir un *guide dactylographié* pour l'installation des mises à jour.

1.7. Apparence et perception : ergonomie et convivialité de la solution

Cette partie contient toutes les exigences relatives à l'ergonomie et à l'utilisation du logiciel. L'ergonomie consiste à améliorer la convivialité, la lisibilité, l'efficacité (rapidité d'encodage ou encore temps de réponse et transparence), l'adéquation par rapport à l'utilisateur et la clarté d'une application.

1.7.1. Interface graphique

Cette section décrit les exigences concernant l'esprit de l'interface. Il ne s'agit pas de donner les contraintes sur la façon dont les interfaces sont programmées, ni d'en spécifier leurs contenus. Ces exigences traitent du style d'écriture, des couleurs à utiliser, du degré d'interaction, etc.

Exigences types

L'interface doit utiliser un ensemble $X_{1..N}$ d'attributs graphiques.

Où X est un attribut graphique (style et taille d'écriture, couleur, etc.).

Cette exigence peut aussi s'exprimer de plusieurs manières, par exemple:

- Les interfaces seront *aux couleurs de la société.*

Le degré d'interaction entre l'utilisateur et les interfaces doit se faire selon une série $X_{1..N}$ de dispositions.

Où X est une disposition à prendre pour augmenter le degré d'interaction (environnement textuel ou graphique, langue souhaitée, touche de type raccourcie, selon l'environnement des utilisateurs).

L'environnement des utilisateurs détermine certaines contraintes graphiques : si le produit est destiné à une aide en ligne, si celui-ci s'adresse à une catégorie d'âge, si l'utilisation du logiciel se fait dans des endroits très lumineux, etc.

- Le degré d'interaction entre l'utilisateur et les interfaces doit se faire selon les dispositions suivantes :
  - *colorées et très attrayantes pour des adolescents.*
  - *la langue d'interface est le français.*

1.7.2. Le style du produit

Cette section concerne les exigences décrivant les caractéristiques d'accroche du produit, c'est-à-dire comment le produit sera perçu par ses futurs utilisateurs.

Exigence type

L'interface doit utiliser un ensemble $X_{1..N}$ de styles caractéristiques.

Où X est un style permettant d'établir un contexte d'environnement de l'entreprise.

On peut distinguer deux grandes familles de style :

L'apparence : classique, moderne, conservatrice, institutionnelle, très colorée et destinée à des enfants (préciser l'âge), etc.
La perception : suscite le professionnalisme, la confiance, la réaction, l'achat (en ligne), le déplacement, etc.

- L'interface utilisera *un style contemporain.*

1.7.3. Facilité d'utilisation et aide

Exigences types

Le système doit pouvoir être utilisé pour une certaine catégorie X d'utilisateurs ayant suivi au maximum des type $Y_{..N}$ d'adaptation pour l'utilisation.

Où X est une catégorie d'utilisateurs (selon l'âge, le profil (compétent ou non), etc.).
Y est une adaptation nécessaire pour l'utilisation du logiciel (formation, aucune formation, temps d'adaptation, etc.).

- Le système doit pouvoir être utilisé par des *utilisateurs de compétence moyenne* n'ayant suivi *aucune formation.*

Le système doit guider l'utilisateur, avec un ensemble X de mécanismes, afin d'éviter qu'il ne fasse des erreurs.

Où X est mécanisme pour guider l'utilisateur pendant l'utilisation du logiciel (par un ensemble de rappels de certaines règles à respecter, par une confirmation de l'action, par une vérification des valeurs possibles de certains paramètres, etc.).

- Le système doit guider l'utilisateur, *à l'aide d'une bulle d'aide*, pour éviter qu'il ne fasse des erreurs.

Le système doit être accompagné d'un ensemble $X_{1..N}$ de supports, accessibles à des temps $Y_{1..N}$ pour l'aide à l'utilisateur.

Où X est un type de support (manuel d'utilisation, aide interactive, tutorial, help desk, aide en ligne, etc.).
Y est une période de temps (aucune, à certaines heures, selon les jours, etc.).

- Le logiciel doit être accompagné *d'une hot-line accessible de 6H00 à 22H00* aider l'utilisateur.
- Le logiciel doit être fourni *avec un tutorial.*

1.8. Interfaçage de données d'un logiciel à l'autre

Cette section décrit les exigences spécifiant l'ensemble des applications et de leurs données avec lesquelles le produit devra s'interfacer.

Exigences types

Le système doit s'interfacer avec [un ensemble $W_{1..N}$ de données d'] un ensemble $X_{1..N}$ d'application [à une fréquence Y] [via un médium Z utilisé pour l'interfaçage].

Où W est une donnée de l'application X (pour chaque donnée, indiquer si possible son volume et son format).
X est une application (une autre application, une ancienne version du logiciel à acquérir).
Y est la fréquence à laquelle l'interfaçage aura lieu.
Z est un médium d'interfaçage (réseau, etc.).

- Le logiciel doit pouvoir s'interfacer *aux données (dont le volume est 5 mégas) de comptabilité d'une ancienne version de ce logiciel tous les jours, via le réseau Ethernet de la société.*

Le système doit fournir des standards $X_{1..N}$ d'échange de données.

Où X est un standard d'échange de données (XML, OLE, compatible Excel ou autres compatibilités).

1.9. Intégration dans la nouvelle base de données

Exigences types

La base de données doit pouvoir intégrer l'ensemble de données $X_{1..N}$.

Où X est une description de la donnée à intégrer (pour chaque donnée, indiquer si possible son volume et son format).

- La base de données doit pouvoir intégrer avec *la base de données Excel client qui compte 2000 entrées.*

X est chargé de la transposition de l'ensemble $Y_{1..N}$ des données actuelles [en utilisant pour ce transfert une méthode Z de transposition].

Où X est la personne à charge de la transposition (client, fournisseur, etc.).
Y est une description de données à intégrer (pour chaque donnée, indiquer si possible son volume et son format).
Z est une méthode de transposition (à la main, par des requêtes SQL spécifiques, etc.)

- *Le fournisseur* est chargé de la transposition de l'ensemble $Y_{1..N}$ des données actuelles en utilisant pour ce transfert *un encodage des données à la main.*

1.10. Exigences culturelles et politiques

Cette partie contient l'ensemble des exigences spécifiques aux facteurs sociologiques et politiques qui affectent l'acceptabilité du produit.

Exigences types

Le système doit être en accord avec un ensemble $X_{1..N}$ d'exigences culturelles.

Où X est une donnée culturelle (jours fériés des pays européens, liste des pays, système de numérotations de routes belges et français, superstitions, habitudes différentes, autres normes culturelles.).

Le système doit répondre avec un ensemble $X_{1..N}$ d'exigences politiques internes.

Où X est une exigence politique (politique interne à l'entreprise comme l'obligation d'acheter le produit uniquement à une certaine société, etc.).

- Le système doit répondre aux politiques internes suivantes :
  - L'acquisition ne se fait que si la sécurité du logiciel bénéficie d'une certification européenne.
  - L'acquisition ne se fait que si les développeurs internes sont capables de réaliser la maintenance, etc.

1.11. Contraintes légales, contractuelles et normes

Cette section doit contenir toutes les exigences concernant les normes ou règlements auxquels le logiciel doit répondre. Ces règles ou normes peuvent être générales (directives européennes, droit national, etc.).

1.11.1. Exigences légales

Exigence type

Le système doit être en conformité avec la loi X.

Où X est une loi sous laquelle l'entièreté ou une partie du système tombe (droit d'auteur, vie privée, etc.).

- Le système doit être en conformité avec loi du 2 août 2002 dite "sur la vie privée" du Luxembourg.

1.11.2. Normes

Ensemble des exigences spécifiant les standards que le logiciel doit respecter. Ces standards sont des standards de développement, des standards de données et bases de données, des standards d'activité métier, etc.

Exigence type

Le système doit être conforme à un ensemble $X_{1..N}$ de standards.

Où X est une standard (certification Iso 15408, certification IEEE 830, ODBC, MAPI, des standards d'activité métier, etc.).

Conclusion

Ce chapitre représente un état de l'art des exigences non fonctionnelles et est :

- **Utile** car il permet, lors de la création d'un cahier des charges, de ne pas oublier certains types d'exigences. Il peut donc améliorer la qualité du document et en faciliter la rédaction.

- **Novateur** car la mise sous forme d'exigences types est une approche nouvelle. Il permet d'aider à la formulation des exigences non fonctionnelles en les proposant sous une forme plus générique qui pourra être instanciée dans un contexte réel.

Dans cet état de l'art, on peut remarquer que certaines catégories d'exigences non fonctionnelles sont mieux adaptées à la mise sous forme d'exigences types. La sécurité s'y prête particulièrement bien car elle peut facilement se décomposer en sous éléments qui ont été traduis en exigences types. Par contre, pour les exigences légales, aucune découpe n'a été possible car ce domaine est trop vague.

Ce chapitre va être utilisé comme base à la méthodologie exposée au chapitre suivant.

Chapitre 2 : Aide méthodologique à la sélection et à l'instanciation des exigences types

Introduction

Ce chapitre est consacré à la mise en pratique des exigences types définies dans le chapitre précédent. Pour ce faire, nous avons développé une méthodologie qui vise à automatiser partiellement la spécification des exigences non fonctionnelles dans un **domaine** particulier. Par domaine, nous entendons ici un type de logiciel (par exemple : ERP, site Web, CRM,...).

Une fois ce domaine défini, nous avons recherché quels sont les **critères** permettant de décider si une exigence type doit se retrouver ou non dans le cahier des charges. Les critères que nous allons proposer ne sont évidemment pas exhaustifs et peuvent varier selon le domaine choisi.

A ce stade, il est donc possible de savoir si une exigence type doit ou non être reprise dans le cahier des charges. Mais ces exigences sont d'un niveau trop générique que pour être directement introduites. Nous avons donc défini certains **contextes** (par exemple, des fonctionnalités typiques du domaine logiciel choisi) suivant lesquels nous pouvons donner un exemple d'instanciation des exigences types.

Les contextes développés dans ce chapitre ne couvrent évidemment pas l'entièreté du domaine choisi mais permettent néanmoins d'illustrer l'instanciation des exigences types. De plus, pour les exigences non fonctionnelles, il n'est pas toujours possible d'appliquer des 'exigences types' à des contextes précis. C'est pourquoi de nombreuses exigences seront traitées directement au niveau du domaine choisi.

Domaine

Le domaine étudié dans cette partie est celui des ERP (Entreprise Ressource Planning).
Les ERP sont des progiciels de gestion intégrée qui ont vu le jour dans les années 70. Ces logiciels, très largement utilisés, peuvent remplacer la totalité du système d'information d'une entreprise.
Mais quelle est la définition exacte d'un ERP ? Le site CXP[6] (conseils en choix de logiciels et de progiciels) nous expose l'ensemble des caractéristiques pour qu'un progiciel soit considéré comme 'intégré'. Il doit :

- Emaner d'un concepteur unique ;
- Garantir à l'utilisateur l'unicité de l'information, assurée par la disponibilité de l'intégralité de la structure de la base de données à partir de chacun des modules, même pris individuellement ;
- Reposer sur une mise à jour en temps réel des informations modifiées dans tous les modules affectés ;
- Fournir des pistes d'audit basées sur la garantie d'une totale traçabilité des opérations de gestion ;
- Couvrir soit une fonction (ou filière) de gestion, soit la totalité du système d'information de l'entreprise.

[6] http://www.cxp.fr/

L'ERP est donc un système d'intégration informatique de l'entreprise. Il repose sur un "progiciel intégré" **paramétrable** et désigne le système de gestion intégrée étendu à toute l'entreprise.

Ainsi, il intègre toute information saisie et en permet la circulation, la gestion et l'analyse. Etant fondé sur un ensemble unique de base de données, il peut couvrir la majorité des besoins d'affaires de l'entreprise.

Globalement, les intérêts et enjeux [ERP03] de l'implantation d'un ERP au sein d'une entreprise sont :

- Les enjeux économiques :
  - Réduction des coûts par l'automatisation des tâches
  - Identification et quantification des bénéfices
  - Réduction des coûts des matières premières permise par la centralisation des achats
  - Réduction des coûts administratifs des ventes
  - Gain de productivité
- Les enjeux organisationnels :
  - Réduction des délais de traitement ou d'acheminement des flux d'informations dans l'entreprise
  - Amélioration des processus
  - Accroissement de la capacité d'adaptation de l'entreprise

Les bénéfices au niveau du système d'information sont également multiples :

- Stockage des données utiles en un endroit unique. Il existe ainsi une seule base clients pour les fonctions commerciales et de facturation. L'intérêt réside dans la garantie d'absence d'incohérence et la suppression des saisies redondantes.
- Intégration des flux dans l'entreprise étendue, ce qui autorise l'accès à l'ERP par des utilisateurs mobiles et des partenaires externes (via Internet et Extranet).
- Système d'informations en temps réel comme support aux décisions d'affaires. Les problèmes de synchronisation des données disparaissent, ce qui permet une gestion des budgets avec un contrôle en temps réel des engagements.

Le graphique ci-dessous représente l'ensemble des fonctionnalités couvertes par un ERP ainsi que leur relation vis-à-vis des clients/fournisseurs[7].

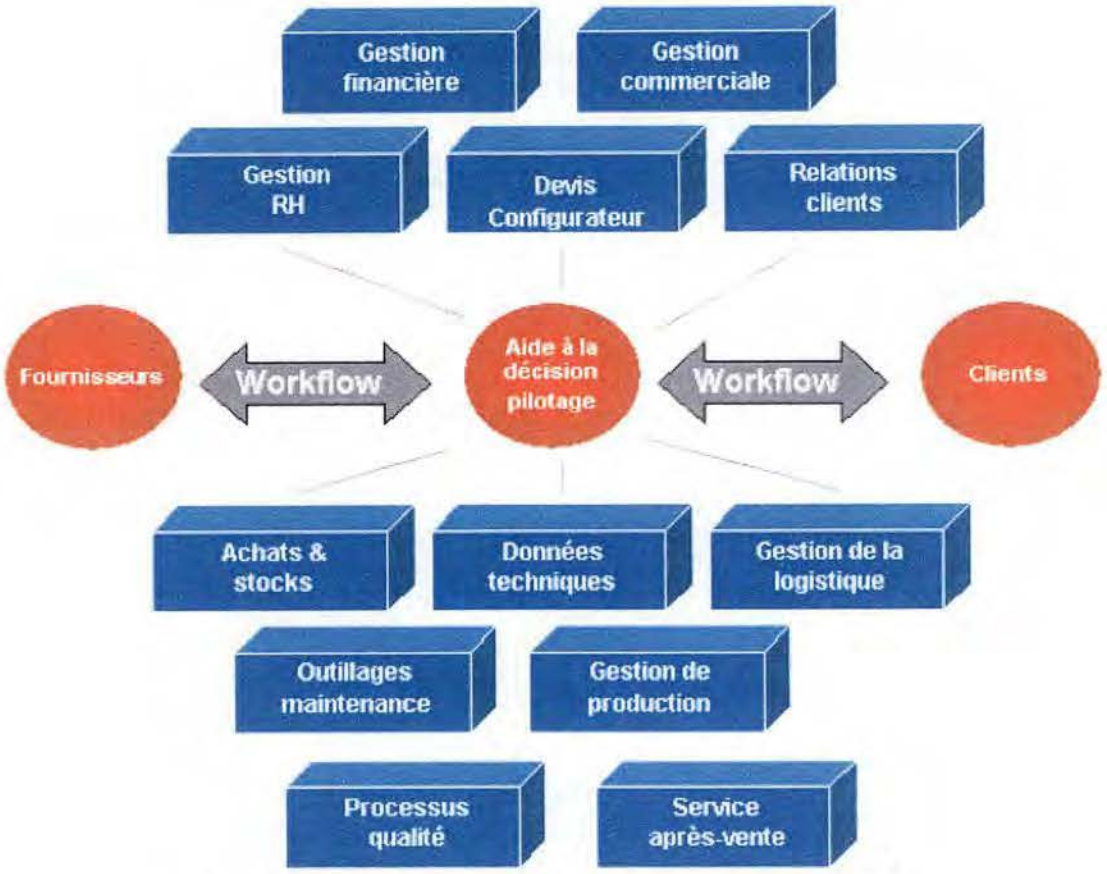


Figure 1 : Ensemble des fonctionnalités couvertes par un ERP

La **gestion financière** par exemple recouvre les domaines de :

- ***comptabilité*** (générale, financière, analytique) ;
- ***gestion des immobilisations*** : Optimisation de la gestion des biens immobilisés depuis l'acquisition jusqu'à la cession en passant par leurs amortissements ;
- ***credit management*** : tous les aspects de la gestion du risque client, permettant de définir la politique de crédit client, d'anticiper, d'organiser et de piloter les actions de recouvrement des encours clients.

La **gestion commerciale** regroupe les :

- ***offres de prix*** : personnalisées au client, remises et promotions ;
- ***prévisions commerciales*** : par produit, à court ou long terme ;
- ***statistiques commerciales*** : par secteur ou produit ;
- ***calculs des prix de revient*** : sur base de données actuelles ou bien par simulation.

[7] source : http://www.alizee-info.fr/solutions.htm

La **gestion des relations clients** facilite la démarche consistant à optimiser les ventes par une meilleure connaissance des besoins clients et prospects et à accroître leur satisfaction par la qualité des prestations de services proposés. Ce module incorpore entre autres des fonctionnalités pour :

- accéder directement aux ***informations pertinentes*** concernant un client ;
- ***planifier*** les relances téléphoniques ou les envois d'e-mail ;
- connaître les ***besoins des clients*** de manière approfondie ;
- visualiser aisément et rapidement un **historique** des derniers contacts avec le client (par exemple, ses dernières commandes, l'éventuelle existence d'un litige avec ce client,...)

La **gestion des achats et stocks** permet de :

- gérer la **liste des fournisseurs** (produits offerts, prix, historique des transactions,etc.) ;
- émettre automatiquement des **ordres de réapprovisionnement** si les stocks franchissent un certain seuil ;
- s'assurer que les **stocks** seront suffisants pour les productions en cours et futures ;
- assurer la **traçabilité** des stocks et ce, même s'ils se trouvent sur plusieurs sites.

La **gestion de la production** comprend :

- le **référencement** de tous les outils de production,
- le **suivi** des ordres de production,
- la **traçabilité** matières premières/produits finis.

L'ERP permet évidemment de stocker toutes les informations techniques concernant les produits (informations très utiles lors de la création de catalogues) mais également de gérer le service après-vente ou d'établir des plans de vérification de la qualité des produits.

Les exemples ci-dessus ne représentent assurément qu'une infime partie des fonctionnalités offertes par un ERP. Néanmoins, ils permettent d'imaginer des possibilités offertes par ce type de logiciel.

Plusieurs éléments nous ont amenés à choisir le domaine des ERP pour illustrer notre méthodologie :

- la large utilisation de ce type de logiciel dans les entreprises de toutes tailles,
- l'importance de l'ERP dans les activités de l'entreprise,
- la nécessité d'adapter l'ERP en fonction des besoins de l'entreprise.

Contextes

Dans la suite de cette partie nous considérons les trois fonctionnalités spécifiques suivantes :

1. La comptabilité analytique

Ce type de comptabilité fait partie du module de gestion financière. Elle a pour but notamment d'analyser les éléments de coûts et de bénéfices afin d'obtenir une analyse de rentabilité par centre de coûts ou par produit. Cette analyse peut être effectuée selon divers critères appelés axes analytiques. C'est donc via cette fonctionnalité qu'il est possible d'étudier le seuil de rentabilité d'un produit, son coût marginal, etc.

2. La planification des ordres de fabrication

Un ordre de fabrication définit quel article doit être traité, à quel endroit, à quel moment et quelle est la charge de travail exigée. Il décrit également quelles ressources doivent être employées et comment les coûts de l'ordre doivent être imputés[8].
La planification de ces ordres fait partie de la gestion de la production. Son but est de planifier à l'avance les productions et l'utilisation des machines afin d'en optimiser la disponibilité. La planification doit donc être basée sur des données constamment mises à jour. Elle doit aussi être facilement adaptable et flexible pour permettre la réalisation de toute nouvelle commande.

3. L'accès aux informations pertinentes des clients

Cette fonctionnalité fait partie de la gestion des relations clients. Lors d'un contact avec un client (par téléphone, mail,...), il est en effet très important que son interlocuteur puisse très rapidement accéder aux informations de ce client. Ces informations peuvent être ses dernières commandes, l'état de son compte client, ses plaintes éventuelles,...

Ces fonctionnalités ont été choisies car elles appartiennent aux différentes fonctions majeures de l'ERP (gestion financière, gestion de la production, gestion des relations clients). En outre, elles ont des besoins en matière d'exigences non fonctionnelles assez variés (sécurité, vitesse, etc.)

Néanmoins, certaines exigences non fonctionnelles ne peuvent pas être traitées et instanciées à un niveau aussi fin des fonctionnalités du domaine traité car elles s'appliquent au système global. Ce type d'exigences sera donc traité directement au niveau du domaine (c'est-à-dire au niveau de l'ensemble de l'ERP).

Comme résultat, ce chapitre contient une structure qui, si elle est appliquée à un cas réel, permet aisément de définir si une exigence type doit être inclue dans le cahier des charges ou non. Cette structure permet donc d'automatiser partiellement la sélection des exigences non fonctionnelles. Elle présente également quelques exemples d'instanciations propres au domaine des ERP ou aux contextes sélectionnés dans ce cadre.

[8] source : http://help.sap.com

2.1. Exigences de sécurité

Deux critères prépondérants interviennent dans la détermination des exigences de sécurité : le nombre d'employés et leur profil. Dans le domaine de la sécurité, ce profil est essentiellement défini par le degré de confiance accordé aux employés.

Complémentarité de ces deux critères :

- Lorsque le nombre d'employés est considéré comme faible, le degré de confiance des employés détermine hautement le niveau de sécurité à utiliser. Ainsi, si les utilisateurs sont des personnes de confiance, un niveau de sécurité moins élevé sera suffisant.

- Lorsque le nombre d'employés est considéré comme moyen, les deux critères déterminent de manière complémentaire le niveau de sécurité.

- Lorsque le nombre d'employés est considéré comme fort, le critère du profil est moins important puisqu'on ne peut l'estimer pour l'ensemble des utilisateurs.

Ces deux critères seront adaptés et complétés par d'autres si nécessaire.

2.1.1. Contrôle d'accès au système

Cette section doit être traitée au niveau de l'**ERP** car elle concerne l'accès au système (la session).

Les critères déterminants sont **le degré de confidentialité des informations, le nombre d'utilisateurs et leur profil.** En effet, plus le degré de confidentialité et le nombre d'utilisateurs augmentent, plus il est intéressant de rendre le système inaccessible à toutes personnes non autorisées.

- Lorsque le nombre d'utilisateurs est faible, le degré de confiance élevé et le degré de confidentialité des informations du système faible, il n'est pas nécessaire, hormis demande explicite, de spécifier les exigences types de ce point. En effet, il est inutile de protéger l'accès à des données qui ne sont pas importantes.

  Pour un nombre important d'utilisateurs , il peut s'avérer nécessaire d'établir un accès au système pour permettre la gestion des utilisateurs ou de groupes d'utilisateurs. Dans ce cas, il suffit de reprendre les exigences du paragraphe suivant.

- Quel que soit le nombre d'utilisateurs, si le degré de confidentialité des informations du système est moyen, il est alors nécessaire de spécifier les exigences types suivantes afin de limiter l'accès au système :

  - Le système doit limiter le domaine des attributs de sécurité de la session d'un utilisateur en fonction des attributs de sécurité $X_{1..N}$. de celui-ci.

    Exemple :
    L'ERP doit limiter le domaine des attributs de sécurité de la session d'un utilisateur en fonction *des droits d'accès de cet utilisateur à ces informations.*

- Le système doit être capable de refuser l'ouverture d'une session en fonction d'un ensemble $X_{1..N}$ d'attributs.

  Exemple :
  L'ERP doit être capable de refuser l'ouverture d'une session *si l'utilisateur se trompe de mot de passe.*

– Si le nombre d'utilisateurs est moyen ou élevé et si le degré de confidentialité des informations du système est élevé (conditions les plus probables pour un ERP), il est dès lors nécessaire de spécifier, en plus des exigences précédentes, les exigences types suivantes :

  - Le système doit pouvoir gérer un nombre N de sessions parallèles pour un même utilisateur.

  - Dans des circonstances $X_{1..N}$, le système doit verrouiller la session de l'utilisateur en effectuant une série $Y_{1..N}$ de mesures rendant l'accès aux données de l'utilisateur impossible sans un déverrouillage.

    Exemple :
    Après un certain temps N d'inactivité, l'ERP doit verrouiller la session de l'utilisateur *en effaçant le contenu de l'écran d'affichage et en désactivant tout moyen d'accès à cette session sans qu'elle ne soit déverrouillée.*

  - Le déverrouillage consiste en une série $X_{1..N}$ d'opérations successives.

    Exemple :
    Le déverrouillage consiste *à demander le mot de passe utilisateur, puis à reconstituer l'écran d'affichage tel qu'il était avant le verrouillage.*

  - Le système doit terminer une session à la suite des $X_{1..N}$ événements.

    Exemple :
    L'ERP doit terminer une session *soit après un temps N de verrouillage, soit sur demande de l'utilisateur.*

Ces exigences permettent donc de limiter au maximum les risques d'accès non autorisés à une session, ce qui est très primordial pour la protection des données sensibles.

2.1.2. Identification, authentification

2.1.2.1. Identification

Dans le cas d'une fonctionnalité critique pour l'entreprise, il est nécessaire de pouvoir établir les responsabilités en cas d'action erronée ou malveillante.
L'identification permet aussi de connaître les droits de l'utilisateur et de déterminer, en fonction de ceux-ci, l'accès aux données et fonctionnalités (voir Authentification, point 2.1.2.2.).

Cet élément est traité au niveau de l'**ERP** car l'identification ne se fait que pour l'accès à la session.

Le critère déterminant pour ce point est **l'existence de sessions d'utilisateurs**.

- S'il existe des sessions d'utilisateurs, alors il n'est pas nécessaire de spécifier l'exigence type de ce point.

- S'il n'existe pas de sessions d'utilisateurs, alors il est nécessaire de spécifier l'exigence type de la manière suivante :

    - Le système doit exiger que chaque utilisateur soit identifié avec succès avant d'autoriser toute action sur des fonctionnalités ou des données du système.

    Dans ce cas, l'identification ne s'opérera qu'une seule fois (avant l'ouverture de session).

2.1.2.2. Authentification

Le mécanisme d'identification précède toujours une authentification lors de l'ouverture de session. Par contre, si l'identification n'a lieu qu'une seule fois, l'authentification doit avoir lieu lors de chaque accès aux ressources et fonctionnalités. En effet, il faut pouvoir déterminer si l'utilisateur dispose des droits d'accès ou privilèges pour pouvoir réaliser l'action demandée au système.

A l'exception de la première exigence, cet aspect est traité au niveau des **fonctionnalités de l'ERP** car certaines fonctionnalités sont plus critiques que d'autres et contiennent des données plus sensibles que d'autres.

Les critères déterminants sont **la criticalité des fonctionnalités et des données**. En effet, plus la criticalité et le nombre d'utilisateurs augmentent, plus il est indispensable de protéger l'accès aux fonctionnalités et aux données.

Voici l'exigence relative au premier accès. Celle-ci est nécessaire si une gestion des sessions d'utilisateurs existe :

- Le système doit exiger que tous les utilisateurs soient authentifiés avec succès, via des mécanismes $X_{1..N}$, avant d'autoriser toute action à des fonctionnalités ou des données du système.

- Si la criticalité des fonctionnalités ou des ressources est faible, alors il n'est pas nécessaire de spécifier les exigences types de ce point.

- Si la criticalité des fonctionnalités ou des ressources est élevée, alors il est nécessaire de spécifier, pour celles-ci, les exigences suivantes :

    - Le système doit autoriser l'accès aux utilisateurs $X_{1..N}$ authentifiés avec succès [via des mécanismes $Y_{1..N}$,] avant d'autoriser tout accès à des fonctionnalités ou des données $Z_{1..N}$ du système.

        Exemple pour la comptabilité analytique :
        Le système doit autoriser l'accès *aux responsables de la comptabilité* authentifiés avec succès via des mécanismes de vérification *des droits d'accès*, avant d'autoriser tout accès à *la fonctionnalité de la comptabilité analytique.*

- En cas d'échec X, le système effectue une action Y.

  Exemple pour la comptabilité analytique:
  En cas d'accès non autorisé au module de la comptabilité analytique, le logiciel *terminera la session de l'utilisateur.*

  Exemple pour la comptabilité analytique:
  En cas de tentative d'accès non autorisé aux données sensibles du module de la comptabilité analytique, le logiciel *enverra un message de refus d'accès et enregistrera dans la trace d'audit l'événement correspondant.*

- Selon les conditions $X_{1..N}$, le nombre de tentatives d'authentification infructueuses est limité à N essais.

  Exemple pour la comptabilité analytique :
  Dans le cas d'un accès aux fonctionnalités de la comptabilité analytique, le nombre de tentatives infructueuses est limité à *N* essais.

- Selon les conditions $X_{1..N}$, le temps acceptable pour l'opération d'authentification est de maximum N secondes.

  Exemple pour le planning des ordres de fabrication:
  Dans le cas d'un accès au planning des ordres de fabrication, le temps acceptable pour l'opération d'identification *est de maximum N secondes.*

- Le système doit être capable de détecter et d'empêcher l'utilisation de données d'authentification qui ont été contrefaites ou copiées.

2.1.3. Confidentialité du Système

2.1.3.1. Cryptage

Cette section permet de protéger toutes les informations relatives aux données stratégiques, comme les coûts de fabrication, les coûts marginaux, le résultat des études pour la diminution des coûts, l'augmentation des profits, etc. Autant de données confidentielles et stratégiques à rendre illisibles de l'extérieur, voire de l'intérieur du système.

Ce point est traité au niveau des **fonctionnalités de l'ERP** car les données relatives à certaines fonctionnalités sont plus critiques que d'autres.

Les critères déterminants sont **la criticalité des données.** En effet, plus les données sont importantes, plus il est indispensable de mettre en place des mécanismes de cryptage.

- Si la criticalité des données est faible, il n'est pas nécessaire, sauf sur demande explicite, de spécifier l'exigence type de cette section. En effet, des mécanismes de cryptage de données non sensibles pourraient alourdir le système.

- Si la criticalité des données est forte, alors il convient de spécifier l'exigence type suivante :

- Le cryptage de type W est utilisé pour l'ensemble des données $X_{1..N}$ [de types Y] [lors d'une action Z]

  Exemple pour la comptabilité analytique:
  Le cryptage *asymétrique conforme à la norme X.509* est utilisé *pour la communication des données qualifiées de sensibles des fonctionnalités de la comptabilité analytique.*

  Exemple pour l'accès aux informations pertinentes des clients:
  Le cryptage *asymétrique conforme à la norme X.509* est utilisé pour *le stockage de l'ensemble des informations pertinentes des clients.*

2.1.3.2. Anonymat, pseudonyme, non liabilité, non observabilité

Pour simplifier la lecture, ces quatre types d'exigences sont groupées sous le terme de confidentialité.

Les données dans ce point sont réduites aux données des utilisateurs.

L'exigence type relative à la décentralisation des données relatives à la vie privée des utilisateurs ne doit pas être inclue dans ce point. En effet, un des principes de l'ERP est le stockage des données utiles en un seul endroit (voir introduction de ce chapitre).

Ce point est traité, d'une part, au niveau des **fonctionnalités de l'ERP** car la confidentialité des utilisateurs dépend de certaines fonctionnalités et d'autre part, au niveau de l'ERP, car certaines exigences ont une portée générale sur logiciel.

Les critères déterminants sont **la criticalité des fonctionnalités et des données** et **le nombre d'utilisateurs et leurs profils.** En effet, plus ces critères sont importants, plus il est indispensable de protéger la confidentialité des utilisateurs.

- Si le nombre d'utilisateurs est faible, si le degré de confiance est élevé et si la criticalité de certaines fonctionnalités est faible, alors il n'est pas nécessaire, sauf sur demande explicite, de spécifier les exigences types pour ces fonctionnalités. En effet, les informations ne sont pas critiques pour l'activité de l'entreprise, il n'est donc pas nécessaire de les intégrer dans cette section.

- Si le nombre d'utilisateurs est faible, si le degré de confiance est élevé et si pour certaines fonctions la criticalité des ressources est forte, alors il n'est pas nécessaire, sauf sur demande explicite, de spécifier les exigences pour ces fonctionnalités. En effet, la confidentialité doit garantir la protection des utilisateurs contre la découverte et le mauvais usage de son identité d'utilisateur. Les exigences types de ce point ne sont donc pas spécialement utiles si le nombre d'utilisateurs est faible.

- Si le nombre d'utilisateurs est moyen ou fort, si la criticalité de certaines fonctionnalités et données est faible, alors il est nécessaire pour ces fonctionnalités de spécifier l'exigence type suivante :

  - Le système doit garantir qu'un ensemble $X_{1..N}$ d'utilisateurs soit incapable de déterminer la véritable identité d'un utilisateur [lorsque celui-ci utilise une ressource ou une fonction/service].

Exemple pour la comptabilité analytique:
Le logiciel doit garantir que *tous les simples utilisateurs* soient incapables de déterminer la véritable identité d'un utilisateur *lorsque celui-ci exécute les fonctionnalités de la comptabilité analytique.*

En effet, si le nombre d'utilisateurs est élevé, les exigences générales relatives à la confidentialité s'appliquent pour l'ensemble de l'ERP. Dès lors, il faut pouvoir aussi spécifier les fonctionnalités ne nécessitant pas de connaître la véritable identification des utilisateurs.

- Si le nombre d'utilisateurs est moyen ou fort, si la criticalité de certaines fonctionnalités est forte, il alors est nécessaire, pour ces fonctionnalités, de spécifier les exigences types suivantes:

  - Le système doit garantir qu'un ensemble $X_{1..N}$ d'utilisateurs soit incapable de déterminer la véritable identité d'un utilisateur à partir d'un pseudonyme [lorsque celui-ci utilise une donnée ou une fonction/service].

    Exemple pour la comptabilité analytique :
    L'ERP doit garantir que *tous les utilisateurs qualifiés comme simples utilisateurs* soient incapables de déterminer le véritable nom d'un utilisateur à partir d'un pseudonyme *lorsque celui-ci exécute une fonctionnalité de calcul des coûts de production anticipés.*

  - Le système doit fournir à ensemble $X_{1..N}$ d'utilisateurs, la possibilité de déterminer la véritable identité d'un utilisateur à partir d'un [ensemble $Y_{1..N}$ d'] alias associé à celui-ci [et uniquement sous les conditions $Z_{1..N}$].

    Exemple pour l'accès aux informations pertinentes des clients :
    Le logiciel doit fournir aux *gestionnaires du réseau,* la possibilité de déterminer la véritable identité d'un utilisateur à partir *d'un ensemble d'alias* associés à celui-ci *si l'enregistrement d'un événement suspect d'audit est réalisé lors de l'utilisation d'une fonctionnalité d'accès aux informations pertinentes des clients.*

  - Le système doit pouvoir créer un alias, accepter l'alias de l'utilisateur et contrôler sa conformité à la métrique [(X)] utilisée pour les alias.

    Exemple :
    L'ERP doit pouvoir créer un alias, accepter l'alias de l'utilisateur et contrôler sa conformité à la métrique *utilisée pour les alias.*

  - Sous les conditions $X_{1..N}$, le système doit pouvoir fournir à l'utilisateur un alias identique à un précédent. Dans les autres cas, le système fournira un alias sans relation avec les précédents alias.

    Exemple pour la comptabilité analytique:
    Lors d'un accès aux données qualifiées de critiques pour des fonctionnalités de la comptabilité analytique et enregistrées au nom de l'alias et non au véritable nom de l'utilisateur, l'ERP doit pouvoir fournir à l'utilisateur un

alias identique à un précédent. Dans les autres cas, le système fournira un alias sans relation avec les précédents alias.

- Le système doit garantir qu'un ensemble $X_{1..N}$ d'utilisateurs ne peut faire le lien entre l'utilisation répétée d'ensemble $Y_{1..N}$ de fonctions/services et l'utilisateur.

  Exemple pour la comptabilité analytique:
  L'ERP doit garantir *qu'un utilisateur qualifié comme simple utilisateur* ne peut faire le lien entre l'utilisation répétée *d'une fonction relative à la comptabilité analytique* et l'utilisateur.

- Le système doit garantir qu'un ensemble $X_{1..N}$ d'utilisateurs ne peut pas observer l'utilisation d'un ensemble $Y_{1..N}$ de ressources ou de fonctions/services d'un autre ensemble $Z_{1..N}$ d'utilisateurs.

  Exemple pour la comptabilité analytique :
  L'ERP doit garantir que *les utilisateurs qualifiés comme simples utilisateurs* ne peuvent pas observer *l'utilisation des données des coûts futurs de production d'autres utilisateurs.*

- Le système doit fournir à un ensemble $X_{1..N}$ d'utilisateurs autorisés la possibilité d'observer l'utilisation d'un ensemble $Y_{1..N}$ de ressources ou de fonctions/services.

  Exemple pour la comptabilité analytique:
  L'ERP doit fournir *aux utilisateurs qualifiés comme administrateurs réseaux autorisés* la possibilité d'observer l'utilisation *des fonctionnalités critiques de la comptabilité analytique.*

Ces exigences constituent l'ensemble des mesures à prendre pour la protection de la vie privée. Elles sont nécessaires car le nombre d'utilisateurs le justifie. Elles sont suffisantes car leur respect est une garantie de protection de la vie privée.

Lorsque le nombre d'utilisateurs est suffisamment important, il est conseillé de faire appel à un consultant spécialisé pour la rédaction de cette partie.

2.1.4. Gestion de la Sécurité

2.1.4.1. Gestion des fonctions et des données du système

Cette partie permet à des utilisateurs autorisés de contrôler la gestion des données et des fonctions du système relatives à la sécurité telles que les fonctions d'audit ou encore d'identification et d'authentification.

Ce point est traité au niveau de l'**ERP** car il s'agit d'une gestion pour l'ensemble du système.

Le critère déterminant est **le besoin de paramétrage du système par des utilisateurs internes autorisés.**

- Si ce besoin est faible, alors il ne sert à rien de spécifier les exigences types de ce point (par exemple, le paramétrage se fera par des personnes externes).

- Si ce besoin est élevé, alors il est nécessaire de spécifier les exigences types suivantes :

  - Le système doit fournir des fonctionnalités pour permettre à un ensemble $W_{1..N}$ d'utilisateurs de pouvoir faire des modifications $X_{1..N}$ du comportement des fonctionnalités $Y_{1..N}$ [possédant un ensemble $Z_{1..N}$ de règles ou conditions de fonctionnement modifiables].

    Exemple :
    Le logiciel doit fournir des fonctionnalités pour permettre *à un administrateur autorisé* de pouvoir *modifier les règles du contrôle du flux d'informations.*

    Le logiciel doit fournir des fonctionnalités pour permettre à *un administrateur autorisé* de pouvoir *modifier le degré d'audit exercé sur un utilisateur ou sur certaines fonctionnalités.*

  - Le système doit fournir des fonctionnalités pour permettre à un ensemble $X_{1..N}$ d'utilisateurs de modifier un ensemble $Y_{1..N}$ d'attributs des fichiers du système et des utilisateurs.

    Exemple :
    Le système doit fournir des fonctionnalités pour permettre *au propriétaire d'une donnée* de modifier *l'attribut de confidentialité de la donnée.*

2.1.4.2. Gestion des utilisateurs

La partie gestion des utilisateurs et groupes d'utilisateurs permet la gestion des comptes utilisateurs. Les exigences de cette section seront reprises à condition d'avoir spécifié les exigences types du «contrôle d'accès au système».

Ce point est traité au niveau de l'**ERP** car la gestion des utilisateurs s'applique à l'ensemble de l'ERP.

Le critère déterminant est **l'accès au système**. En effet, si l'accès ne nécessite pas d'ouverture de session, il n'y a pas de gestion des utilisateurs. Par contre, certains systèmes pourraient avoir une gestion des utilisateurs pour l'accès à certaines fonctionnalités. Dans ce cas, seule la première exigence type doit être précisée.

- Si l'accès au système ne se fait pas par l'établissement de sessions, alors il n'est pas nécessaire de spécifier les exigences types de ce point.

- Si l'accès au système se fait par l'établissement de sessions, alors il faut spécifier dans ce point les exigences types suivantes :

  - Le système doit fournir une série $X_{1..N}$ de fonctionnalités pour la gestion des comptes utilisateurs.

    Exemple :
    Le logiciel doit fournir des fonctionnalités *pour la création, la modification, la suppression, le blocage et déblocage des comptes, ainsi que pour l'historique d'accès etc.*, pour la gestion des comptes utilisateurs

- Le système doit fournir une série $X_{1..N}$ de fonctionnalités pour la gestion des groupes d'utilisateurs

  Exemple :
  Le logiciel doit fournir des fonctionnalités *pour la création, la suppression, la modification (ajout, modification, suppression d'un membre) des groupes* pour la gestion des groupes d'utilisateurs.

2.1.4.3. Gestion des attributs de sécurité

La section gestion des attributs permet essentiellement (dans le cadre des fonctionnalités d'un ERP) de définir les attributs nécessaires d'un utilisateur (en terme de droits d'accès) pour avoir accès aux différentes données ou fonctionnalités.

Ce point est traité au niveau de l'**ERP** car la gestion des attributs de sécurité s'applique à l'ensemble de l'ERP.

Le critère déterminant est **l'existence de mécanismes d'authentification basés sur des attributs.** En effet, si l'on décide d'opter pour des mécanismes d'identification et d'authentification pour certaines fonctionnalités, il doit être possible de modifier les attributs pour ces mécanismes afin d'inclure et d'exclure des accès.

– S'il existe un mécanisme d'authentification basé sur des attributs, il est nécessaire de spécifier dans cette partie les exigences suivantes :

  - Le système doit permettre à un ensemble $X_{1..N}$ d'utilisateurs d'effectuer un ensemble d'actions $Y_{1..N}$ [sur] les attributs de sécurité d'un utilisateur.

    Exemple :
    Le logiciel doit permettre *aux administrateurs système* de *consulter, modifier, réinitialiser à la valeur par défaut, supprimer* les attributs de sécurité d'un utilisateur.

  - Lors de l'ajout d'un utilisateur [parmi un ensemble $X_{1..N}$ d'utilisateurs], chaque attribut de sécurité du nouvel utilisateur possède une valeur par défaut [propre à l'ensemble $X_{1..N}$].

    Exemple :
    Lors de l'ajout d'un utilisateur *qualifié d'utilisateur simple*, chaque attribut de sécurité du nouvel utilisateur possède une valeur par défaut.

  - Le système doit pouvoir permettre à un ensemble $X_{1..N}$ d'utilisateurs d'effectuer un ensemble d'actions $Y_{1..N}$ sur les valeurs par défaut des attributs de sécurité.

    Exemple :
    Le logiciel doit pouvoir permettre *aux administrateurs système de consulter, de modifier* la valeur par défaut des attributs de sécurité.

  - Le système doit contenir un ensemble $X_{1..N}$ de valeurs possibles [(restrictives, permissives, ou autres propriétés)] pour les attributs de sécurité, et doit garantir qu'aucun utilisateur ne puisse y déroger.

Exemple :
Le logiciel doit contenir un *ensemble de valeurs possibles (par exemple par une syntaxe composée de N chiffres)* pour les attributs de sécurité, et doit garantir qu'aucun utilisateur n'ait la possibilité d'y déroger.

Les deux premières exigences permettent de spécifier les actions sur les attributs et leurs valeurs par défaut. On considère ici toutes les fonctionnalités (et pas seulement les fonctionnalités critiques) car les droits d'accès sur cette fonctionnalité peuvent changer (par exemple, si l'on souhaite par la suite limiter un accès).

2.1.5. Intégrité des fichiers

2.1.5.1. Backup, imports et exports

La partie sauvegarde des données est nécessaire pour garantir l'intégrité et la pertinence des données (disponibilité des données les plus récentes) en cas de détection d'erreur, de panne ou autre panne/désastre.

Pour les imports et exports, ce point est traité au niveau de l'**ERP** car les contrôles d'accès et du flux des données s'appliquent à l'ensemble des données importées et exportées.

Le critère déterminant est **l'existence des contrôles d'accès (authentification) et du flux d'informations.**

Exigences types pour l'import et l'export :

- Ces exigences se retrouvent toujours dans le cahier des charges si les contrôles d'accès (authentification) et du flux d'informations ont été spécifiés.
    - Le système doit appliquer les contrôles d'accès et du flux d'informations lors de l'exportation de données de l'utilisateur contrôlées par le système, vers l'extérieur du système.
    - Le système doit exporter les données de l'utilisateur avec les attributs de sécurité qui leur sont associés.

Pour les backups, ce point est abordé au niveau des **fonctionnalités de l'ERP** car la sauvegarde dépend de l'importance des données.

Le critère déterminant est **la criticalité d'un ensemble de données d'une fonctionnalité** car il est essentiel de pouvoir privilégier certaines données selon la criticalité des fonctions.

Exigences types pour la sauvegarde:

- Si la criticalité d'un ensemble de données est faible, alors il n'est pas nécessaire de spécifier l'exigence type car leurs sauvegardes ne sont pas importantes pour le bon fonctionnement de l'entreprise.

- Si la criticalité d'un ensemble de données est moyenne, alors cette exigence type doit se toujours se retrouver dans le cahier des charges, avec néanmoins des adaptations pour le temps et l'ensemble des utilisateurs :

  L'unité de temps doit être suffisante pour laisser la priorité aux données les plus critiques. Il n'est pas nécessaire de spécifier l'ensemble des utilisateurs.

- Si la criticalité d'un ensemble de données est élevée, alors cette exigence type doit toujours être reprise dans le cahier des charges, avec néanmoins des adaptations en fonction du temps et de l'ensemble des utilisateurs :

  L'unité de temps doit être minimum. Il est nécessaire de donner une priorité à l'ensemble des utilisateurs utilisant le plus de données sensibles.

  - [Tous les W périodes de temps,] le système doit sauvegarder sur un support de stockage de type X, les informations [appartenant à l'ensemble $Y_{1..N}$ d'utilisateurs ou de fonctionnalités] relatives aux données de type Z [sans devoir arrêter l'exploitation des autres applications].

    Exemple pour la comptabilité analytique:
    Toutes les 24 heures, le système doit sauvegarder sur un support *optique*, les informations *des utilisateurs* relatives aux *données sensibles de la comptabilité analytique.*

    Exemple pour la gestion des informations pertinentes des clients:
    Toutes les 3 heures, le système doit sauvegarder sur un support *magnétique*, *les informations pertinentes des clients.*

2.1.5.2. Annulation

La partie annulation consiste à revenir, suite à une erreur commise, à un point antérieur pour éviter que l'erreur ne se répercute/propage.

Ce point est traité au niveau des **fonctionnalités de l'ERP** car l'annulation dépend de la complexité des fonctions.

Les critères déterminants sont **le profil des utilisateurs** et **la criticalité d'un ensemble de résultats d'une fonctionnalité**. Notons que pour cette sous-section, le profil d'utilisateur sera déterminé par le degré de compétence de l'utilisateur.

Exigences types :

- Le système doit fournir un mécanisme X d'annulation pour l'ensemble des fonctionnalités suivantes : $Y_{1..N}$.

- Le système doit autoriser l'annulation selon l'ensemble $X_{1..N}$ constituant les limites de l'annulation du mécanisme Y.

- Si le degré de compétence est élevé et si la criticalité d'un ensemble de résultats de certaines fonctionnalités est faible, alors il n'est pas nécessaire de spécifier, sauf sur demande explicite, les exigences types de cette section pour ces fonctionnalités. En effet, l'erreur est fort peu probable pour de tels utilisateurs et moins importante car les fonctionnalités sont moins critiques.

- Si le degré compétence est faible et si la criticalité d'un ensemble de résultats de certaines fonctionnalités est faible, alors il peut être nécessaire de spécifier les exigences types de cette section pour ces fonctionnalités. En effet, l'erreur est fort probable pour de tels utilisateurs mais cependant moins importante car les fonctionnalités sont moins critiques. Dans ce cas et s'il existe une demande explicite de disposer de mécanismes d'annulation, on limitera fortement le retour en arrière (par exemple, uniquement de la dernière opération).

  Exemple pour la gestion des informations pertinentes des clients :

  Première exigence :
  Le logiciel doit fournir un *mécanisme d'annulation d'opération pour la gestion des informations pertinentes des clients.*

  Deuxième exigence :
  Le logiciel doit autoriser l'annulation *uniquement de la dernière opération.*

- Si le degré compétence est « α » et si la criticalité d'un ensemble de résultats de certaines fonctionnalités est fort, alors il est nécessaire de spécifier les exigences de cette partie pour ces données. En effet, la fréquence d'erreur dépend du degré « α » de compétence. Il faut dès lors limiter au minimum le retour en arrière (par exemple à N opérations si faible, N-M opérations si fort) selon ce degré « α » de compétence. En effet, même si l'utilisateur est très compétent, personne n'est à l'abri d'une erreur. De plus, l'erreur aurait des répercussions importantes car les fonctionnalités sont critiques. Dès lors, on fixera en plus du degré « α », un seuil S minimum pour le retour en arrière (avec $N-M \geq S$).

  Exemple pour la gestion du planning de planification :

  Première exigence :
  L'ERP doit fournir un *mécanisme d'annulation d'opération pour la gestion du planning de fabrication.*

  Deuxième exigence :
  L'ERP doit autoriser l'annulation *pour les N dernières opérations.*

Ces exigences ont essentiellement pour but de gagner du temps (ne pas devoir tout recommencer) et d'éviter l'exécution irréversible de fonctionnalités critiques ayant des répercussions sur d'autres modules du système. Encore faut-il que l'erreur soit repérée.

2.1.5.3. Politique et fonctions du contrôle du flux d'informations entrant et sortant

La partie politique et fonction de contrôle de flux d'informations entrant et sortant permet le filtrage de l'information garantissant l'intégrité des données lors d'un flux d'information allant de l'extérieur du système vers un utilisateur du système et inversement. Les exigences spécifiées dans cette partie sont importantes car elles déterminent le degré de contrôle et de

filtrage des informations entrantes et sortantes, de détecter la présence de virus, de spams et chevaux de Troie. Ce genre de contrôle est déterminant puisqu'il évite la sortie vers l'extérieur de données qualifiées de confidentielles ou stratégiques et permet de contrôler les informations entrantes pouvant bloquer le système par l'intrusion de mauvais programmes (virus effaçant des données stratégiques, etc.) ou données (fausses données etc..).

Ce point est traité au niveau des **fonctionnalités de l'ERP** car certaines fonctionnalités utilisent plus de données critiques et stratégiques que d'autres.

Les critères déterminants sont **la criticalité d'un ensemble de données d'une fonctionnalité** et **le degré d'ouverture à l'extérieur.** En effet, plus la criticalité des données et le degré d'ouverture sont importants, plus cela représente un risque pour l'activité de l'entreprise car ce sont les utilisateurs qui créent le flux de données critiques. Le degré d'ouverture à l'extérieur détermine l'importance des connexions réseaux établies avec l'extérieur (le nombre de service en ligne est-il important, les services en ligne sont-ils reliés au système, réseau Extranet, etc.).

- Si la criticalité d'un ensemble de données de certaines fonctionnalités est faible et si le degré d'ouverture à l'extérieur est nul, alors il n'est pas nécessaire de spécifier les exigences types de cette section et les règles sur le transfert de données vers l'extérieur, pour ces fonctionnalités. Les données de celles-ci ne présentent aucun risque pour l'entreprise si une personne malveillante parvenait à les exporter. De plus, l'entreprise ne dispose pas de connexions avec l'extérieur.

- Si la criticalité d'un ensemble de données de certaines fonctionnalités est faible et si le degré d'ouverture est moyen ou élevé, alors il n'est pas nécessaire, sauf sur demande explicite, de spécifier dans cette section les exigences types et les règles sur le transfert de données vers l'extérieur, pour ces fonctionnalités. Les données de celles-ci ne présentent en effet aucun risque pour l'entreprise si une personne malveillante parvenait à les exporter. Par contre, il est nécessaire de spécifier les exigences types de cette section, limitées au contrôle des données entrantes car celles-ci peuvent présenter un danger potentiel pour le système.
  Nous retrouvons donc les trois exigences types ci-dessous, mais uniquement pour les données entrantes.

- Si la criticalité d'un ensemble de données de certaines fonctionnalités est forte et si le degré d'ouverture est moyen ou élevé, alors il est nécessaire de spécifier les exigences types de cette section sur le transfert des données aussi bien vers l'extérieur et l'intérieur du système. Dans ce cas-ci, il n'est pas nécessaire de contrôler tous les utilisateurs, seuls ceux manipulant d'importantes données suffiront. En effet, les utilisateurs n'ayant pas accès aux informations sensibles, ne présentent aucun danger pour l'entreprise. Il est aussi nécessaire, comme pour le point précédent, de contrôler les données entrantes.

Exigences types :

- Le système doit appliquer un ensemble $X_{1..N}$ de règles pour le contrôle du flux d'informations à l'ensemble $Y_{1..N}$ d'utilisateurs.

  Exemple pour la comptabilité analytique :
  Le logiciel doit appliquer un *filtrage des données des fonctionnalités de la comptabilité analytique* pour le contrôle du flux d'informations *à l'ensemble des utilisateurs ayant accès à ces fichiers.*

- Dans les conditions $X_{1..N}$, le système doit autoriser le flux d'informations [pour l'ensemble $Y_{1..N}$ d'utilisateurs] selon les règles suivantes : $Z_{1..N}$.

  Exemple pour la gestion des informations pertinentes des clients :
  Lors de communication sécurisée avec une entité extérieure à l'ERP, l'ERP doit autoriser le flux d'informations *de tous les utilisateurs* selon les règles *suivantes : un filtrage des informations sera effectué pour éviter que des données de la gestion des informations pertinentes des clients ne soient échangées sans les droits nécessaires.*

- Le système doit interdire le flux d'informations lorsqu'une règle X est outrepassée [pour l'ensemble $Y_{1..N}$ d'utilisateurs].

  Exemple pour la gestion des informations pertinentes des clients :
  Le logiciel doit interdire le flux d'informations lorsque *le contrôle du flux détecte un risque potentiel d'échange non autorisé des informations pertinentes des clients.*

La première exigence définit les règles à appliquer au contrôle du flux d'informations, et les deux autres, l'autorisation et l'interdiction selon ces règles.

2.1.5.4. Contrôle d'intégrité des données stockées

La partie contrôle des données stockées permet de garantir que les données stockées dans le système (mémoire ou support de stockage) sont correctes. Il s'agit en effet de vérifier les erreurs physiques et d'intégrité pour éviter de les utiliser sans qu'elles ne soient restaurées (cela pourrait provoquer des erreurs d'exécution ou autres, etc.).

Ce point est traité au niveau de l'**ERP** car l'ensemble des données d'un ERP doit être centralisé (principe énoncé dans l'introduction de ce chapitre).

Le critère déterminant est **la criticalité d'un ensemble de données.** En effet, il est plus important de surveiller l'intégrité des données sensibles car de ses informations dépend l'activité de l'entreprise.

- Si la criticalité d'un ensemble de données d'un utilisateur est faible, alors il n'est pas nécessaire, sauf sur demande explicite, de spécifier pour ces données, les exigences types de cette section.

- Si la criticalité d'un ensemble de données d'un utilisateur est élevée, alors il est nécessaire de spécifier pour ces données, les exigences types de cette section. En effet, ces données sont très importantes, il faut donc vérifier leur intégrité en premier.

Exigences types :

- Le système exécute des contrôles d'intégrité de type $X_{1..N}$ pour toutes les données ayant les attributs $Y_{1..N}$.

  Exemple :
  Le logiciel exécute des contrôles d'intégrité de type *checksum* pour toutes les données ayant un *attribut de criticalité élevé.*

- Le système doit effectuer un ensemble $X_{1..N}$ d'actions lorsque celui-ci détecte une erreur d'intégrité.

  Exemple :
  Le logiciel doit *avertir l'utilisateur, s'il en est capable, réparer la donnée, sinon la supprimer avec approbation de l'utilisateur* lorsque celui-ci détecte une erreur d'intégrité.

La première exigence est destinée au contrôle lui-même tandis que la seconde concerne les actions à entreprendre lorsqu'une erreur est détectée.

2.1.5.5. Protection des informations résiduelles

Ce point est traité au niveau des **fonctionnalités de l'ERP** car cette protection doit s'appliquer uniquement pour les données sensibles des fonctionnalités critiques.

Le critère déterminant est **la criticalité des données.** En effet, l'accès aux données critiques effacées mais toujours accessibles physiquement (par exemple, sur un support de sauvegarde) est un danger pour l'entreprise.

- Si la criticalité d'un ensemble de données est faible, alors il n'est pas nécessaire, sauf sur demande explicite, de spécifier l'exigence type de cette section pour ces données. En effet, les informations ne sont pas critiques pour l'activité de l'entreprise. L'accès à celles-ci par des personnes non autorisées (personnes externes ou non) ne représente donc pas un danger important pour l'entreprise. Il n'est donc pas nécessaire de les intégrer dans cette section.

- Si la criticalité d'un ensemble des données est forte, alors il est nécessaire de spécifier l'exigence type de cette section pour ces données. En effet, les informations sont critiques pour l'activité de l'entreprise. L'accès à celles-ci par des personnes non autorisées représente donc un danger important pour l'entreprise.

  Exigence type :

  - Le système doit garantir que l'ensemble des données $X_{1..N}$ sera définitivement inaccessible après effacement.

    Exemple:
    Le système doit garantir que l'ensemble *des données sensibles de l'ERP* sera définitivement inaccessible après effacement.

Ces exigences sont essentielles pour cette partie car l'ERP contient un certain nombre d'informations stratégiques. Il est nécessaire de garantir qu'elles ne soient plus accessibles (même physiquement sur un support) après leur effacement.

2.1.6. Non répudiation des échanges et des transactions

La partie non répudiation des échanges et des transactions permet de fournir des preuves pour les actions et les transactions effectuées. Ces exigences garantissent que le récepteur ne peut contester la réception d'un message et que l'émetteur ne peut contester son envoi.

Ce point est traité au niveau de l'**ERP** car les exigences types de cette section concernent l'importance des données quelle que soit la fonctionnalité d'où elles proviennent.

Les critères déterminants sont **la criticalité des données envoyées, le nombre d'utilisateurs et leur profil** car la non répudiation n'est intéressante que pour les données critiques. Dans cette section, on entend par données critiques les données dont l'envoi avant un certain temps est crucial pour le récepteur.

- Si le nombre d'utilisateurs est faible, si leur profil est fort (degré de confiance élevé), et quelle que soit la criticalité des données ou messages envoyés, alors il n'est pas nécessaire, sauf sur demande explicite, de spécifier les exigences types de cette section. En effet, le nombre d'utilisateurs et leur profil conduiront très rarement à des litiges ; d'autant plus, si les données ne sont pas critiques. Par contre, il peut être nécessaire de spécifier les exigences de cette partie si les données ou les messages sont jugés trop critiques.

- Si le nombre d'utilisateurs est moyen ou élevé, quel que soit leur profil, quelle que soit la criticalité des données ou messages envoyés, il est alors nécessaire de spécifier les exigences types de cette section. En effet, plus le nombre d'utilisateurs augmente, plus le nombre de messages croît lui aussi et plus la nécessité de pouvoir envoyer des messages avec un système de preuve est importante, surtout si le nombre de messages avec un contenu important est élevé. En général, lorsque le nombre d'utilisateurs est moyen ou élevé, le nombre de messages critiques l'est aussi.

Exigences types :

Pour l'émetteur :

- Sur demande de personnes $X_{1..N}$, le système doit pouvoir fournir la preuve de la réception pour l'ensemble $Y_{1..N}$ des informations transmises.

  Exemple :
  Sur demande *de l'envoyeur ou d'une autre personne autorisée*, le système doit pouvoir fournir la preuve de la réception pour des *informations confidentielles* transmises.

Pour le récepteur :

- Sur la demande de personnes $X_{1..N}$, le système doit pouvoir fournir la preuve de l'origine pour l'ensemble $Y_{1..N}$ des informations reçues.

  Exemple :
  Sur la demande *du récepteur ou d'une autre personne autorisée*, le système doit pouvoir fournir la preuve de l'origine pour *des informations confidentielles* reçues.

La première exigence est destinée à fournir une preuve aux émetteurs de messages et la seconde aux récepteurs.

2.1.7. Audit

L'audit concerne toutes les activités d'identification, d'enregistrement, de stockage et d'analyse de l'information liée à la sécurité. Les enregistrements d'audit peuvent être examinés par la suite en vue de détecter les activités touchant à la sécurité et les personnes qui en sont responsables.

Ce point est traité au niveau de l'**ERP** car les exigences types de cette section concernent les mécanismes et fonctionnalités pour l'exécution de l'audit. Dès lors, **la spécification des évènements à auditer** doit se faire, d'une part, pour chaque catégorie et sous catégorie (si elle existe) des exigences non fonctionelles. D'autre part, cette spécification doit aussi se faire dans les exigences fonctionnelles, définissant ainsi les événements à auditer pour les fonctionnalités du logiciel.

Remarquons que dans cette partie, le terme ressource signifie tout ce qui est à auditer (accès aux données, aux fonctionnalités, accès réseaux, etc.).

Les critères déterminants pour l'ensemble de cette partie sont **le degré d'exposition aux dangers venant de l'extérieur, la criticalité des ressources** et **le profil des utilisateurs** car ils déterminent les évènements à auditer.
Le degré d'exposition aux dangers venant de l'extérieur est déterminé par l'importance d'ouverture du système au réseau Internet (le nombre de services en ligne est-il important ?, les services en ligne sont-ils reliés au système ?, etc.).
Ces trois critères déterminent le **degré d'audit :**

- Si le degré d'exposition aux dangers venant de l'extérieur est faible pour certaines ressources, si le profil des utilisateurs est fort (degré de confiance élevé), et si la criticalité de ces mêmes ressources est faible, alors le degré d'audit est nul. En effet, rien ne présente de dangers pour ces ressources.

- Si le degré d'exposition aux dangers venant de l'extérieur est important pour certaines ressources, si le profil des utilisateurs est faible (degré de confiance faible), et si la criticalité de ces mêmes ressources est élevé, alors le degré d'audit sera lui aussi important pour ces ressources. En effet, l'ensemble de ces facteurs constitue un grand risque de violation de ces ressources.

- Si le degré d'exposition aux dangers venant de l'extérieur est faible pour certaines des ressources, si le profil des utilisateurs est élevé (degré de confiance élevé), et si la criticalité de ces mêmes ressources est faible, alors le degré d'audit sera lui aussi faible pour ces ressources. En effet, l'ensemble de ces facteurs ne constitue pas un grand risque de violation de ces ressources du système (car elles ne sont pas intéressantes).

- Si le degré d'exposition aux dangers venant de l'extérieur est faible pour certaines ressources, si le profil des utilisateurs est faible (degré de confiance faible), et si la criticalité de ces mêmes ressources est élevé, alors le degré d'audit sera aussi moyen pour ces ressources. En effet, l'ensemble de ces facteurs peut constituer un grand risque de violation de ces ressources du système mais ce risque est limité aux utilisateurs de l'entreprise.

- Si le degré d'exposition aux dangers venant de l'extérieur est important pour certaines des données et des fonctionnalités, si le profil des utilisateurs est élevé (degré de confiance élevé), et si la criticalité de ces mêmes ressources est faible, alors le degré d'audit sera aussi moyen pour ces ressources. En effet, l'ensemble de ces facteurs ne constitue pas un

grand risque de violations de ces ressources du système car celles-ci sont peu critiques et les utilisateurs sont des personnes de confiance.

Ce **degré d'audit** sera dès à présent le critère déterminant dans le choix des exigences dans chacune des sous-parties qui suivent. Il constitue un attribut de sécurité associé aux attributs des données, des fonctionnalités et des utilisateurs du système.

Quel que soit le nombre d'utilisateurs, l'audit reste l'un des points essentiels pour la sécurité car cette partie écoute tous les évènements ou successions d'évènements potentiellement dangereux pour le système et agit en conséquence. Alors, que l'entreprise soit petite ou grande, si une intrusion est détectée, les conséquences pourraient s'avérer désastreuses, d'autant plus que la Cybercriminalité connaît une croissance démesurée de nos jours[9].

2.1.7.1. Réponse automatique

La partie réponse automatique définit l'ensemble des actions automatiques à entreprendre en cas de détection de violations potentielles du système. En effet, il s'agit d'agir le plus rapidement possible afin d'empêcher que la violation ne devienne réelle.

- Excepté pour un degré d'audit nul pour certaines ressources, il est nécessaire de spécifier dans cette section, l'exigence type suivante :

  - Le système doit effectuer un ensemble $X_{1..N}$ d'actions lors d'une détection de violation potentielle de la sécurité.

    Exemple :
    Le système doit *terminer la session de l'utilisateur responsable* lors d'une détection de violation potentielle de la sécurité.

Le type d'action dépend du type de violation potentielle et donc de l'ampleur des dégâts que la violation engendrerait si elle devenait réelle.

2.1.7.2. Génération de données

La section génération de données décrit les conditions menant à la génération des données lors de la détection d'événements contrôlés par le système. Ces exigences décrivent le niveau d'audit que les évènements possèdent (le nombre d'informations connues sur l'évènement est soit simple, basique, détaillé, non spécifié), le type d'événements qui doit être audité et l'ensemble minimal des informations liées à l'audit qui devrait se retrouver dans les enregistrements d'audit.

Excepté pour un degré d'audit nul pour certaines ressources, il est nécessaire de spécifier dans cette section, les exigences types suivantes :

- Le système doit pouvoir générer un enregistrement d'audit pour les évènements à auditer :
  - De démarrage et d'arrêt des fonctions d'audit,
  - Pour l'ensemble $X_{1..N}$ des évènement à auditer ayant un niveau d'audit Y [ou pour tout autre ensemble $Z_{1..N}$ d'évènements à auditer spécifiquement définis].

[9] www.k-otik.com/news/07.31.audits.php

Exemple :
L'ERP doit pouvoir générer un enregistrement d'audit pour les évènements à auditer:
- de démarrage et d'arrêt des fonctions d'audit,
- *pour tous les évènements* à auditer avec un niveau d'audit *détaillé.*

- Dans chaque enregistrement d'audit, le système doit enregistrer au minimum :
  - la date et l'heure de l'évènement,
  - l'identité de l'utilisateur,
  - le résultat [(succès ou échec)] de l'évènement, le type X d'évènement d'audit,
  - pour chaque type X d'évènement d'audit [(sur base des évènements à auditer contenus dans les composants fonctionnels du système)], l'ensemble $Y_{1..N}$ des informations d'audit pertinentes.

- Le système doit pouvoir relier chaque évènement à auditer avec l'utilisateur responsable de cet évènement.

Il est évident qu'il faut spécifier les informations à mentionner dans les enregistrements d'audit. Il est aussi important de spécifier que le système doit pouvoir connaître l'utilisateur responsable pour pouvoir agir sur celui-ci.

2.1.7.3. Analyse

La section analyse consiste à rechercher les possibles ou réelles menaces contre le système. Cette analyse peut déboucher sur la détection d'intrusion et une série d'actions automatiques en cas de violation imminente de la sécurité. On y définit donc les règles à appliquer pour l'analyse.

Le critère déterminant pour l'ensemble de cette section est **le degré d'audit**.

- Si ce degré est nul pour certaines ressources, alors on ne spécifiera aucune exigence type dans cette section.

- Si ce degré est faible pour certaines ressources, alors on retrouvera, pour ces ressources, les exigences suivantes :

  - Le système doit pouvoir appliquer un ensemble de règles en surveillant les évènements à audités et indiquer, en fonction de celles-ci, une violation potentielle du système.

  - Les règles à appliquer pour la surveillance sont l'accumulation ou la combinaison/succession d'un ensemble $X_{1..N}$ d'évènements à auditer considérés comme des violations potentielles de la sécurité.

    Exemple :
    Les règles à appliquer pour la surveillance sont, pour l'accumulation d'évènements, *une succession d'évènements de tentatives infructueuses puis fructueuses de login effectués à partir d'Internet suivie par une consultation de données sensibles, etc.* considérés comme des violations potentielles de la sécurité.

- Le système doit pouvoir maintenir une représentation en interne de l'ensemble des évènements sélectionnés et de l'ensemble des enchaînements d'évènements qui peuvent indiquer une violation du système.

- Le système doit pouvoir indiquer une violation imminente quand l'activité du système correspond à un évènement ou à un enchaînement d'évènements caractéristiques qui indique une violation potentielle du système.

Il s'agit ici des bases de l'audit, c'est-à-dire, l'ensemble minimum des exigences à retrouver pour l'analyse de l'audit.

– Si ce degré est moyen ou fort pour certaines ressources, alors on retrouvera pour ces ressources, en plus des exigences du paragraphe précédent, les exigences suivantes :

  - Le système doit pouvoir maintenir des profils d'utilisation pour chaque utilisateur. Ces profils individuels représentent un historique des comportements d'un ensemble $X_{1..N}$ d'utilisateurs cibles.

    Exemple :
    Le système doit pouvoir maintenir des profils d'utilisation pour chaque utilisateur. Ces profils individuels représentent un historique des comportements *des utilisateurs ayant accès au module de gestion des ventes.*

  - Le système doit pouvoir maintenir un indice de représentativité pour chaque utilisateur dont l'activité est enregistrée dans un profil. Cet indice indique le degré avec lequel l'activité actuelle de l'utilisateur diffère des modèles d'utilisation représentés dans le profil.

  - Le système doit être capable d'indiquer une violation imminente lorsque l'indice de représentativité dépasse le seuil limite d'une des conditions de l'ensemble $X_{1..N}$ des conditions d'activités anormales.

    Exemple :
    Le système doit être capable d'indiquer une violation imminente lorsque l'indice de représentativité dépasse le seuil limite *du nombre N d'accès aux ressources confidentielles non consultées habituellement.*

Pour le degré d'audit moyen ou fort, nous rajoutons la détection des anomalies basées sur un profil d'utilisateur.

2.1.7.4. Revue d'audit

La section revue d'audit concerne les outils d'audit mis à la disposition des utilisateurs autorisés pour aider à l'examen des données d'audit.
Ces outils sont parfois indispensables, afin, par exemple, de tracer une violation du système non détectée par l'analyse, etc.

- Excepté pour un degré d'audit nul pour certaines ressources, il est nécessaire de spécifier dans cette section, les exigences types suivantes :

    - Le système doit permettre à un ensemble $X_{1..N}$ d'utilisateurs autorisés de lire l'ensemble $Y_{1..N}$ des informations dans les enregistrements d'audit.

        Exemple :
        Le système doit permettre *aux administrateurs autorisés* de lire *toutes les informations* des enregistrements d'audit.

    - Le système doit présenter les informations d'audit d'une façon permettant à l'utilisateur de pouvoir les interpréter.

    - Le système doit pouvoir proposer la possibilité d'effectuer un ensemble $X_{1..N}$ d'opérations sur les données de l'audit selon un ensemble $Y_{1..N}$ de critères liés logiquement à ces opérations.

        Exemple :
        Le système doit pouvoir proposer la possibilité d'effectuer *un tri* sur les données de l'audit selon le *nombre d'occurrences d'un même évènement.*

La première exigence détermine qui a accès au enregistrement, car il est important de garder les règles d'audit hors de portée de tout le monde simplement pour des raisons de sécurité (par exemple, éviter qu'une personne ne puisse détecter une faille du système en découvrant quelles ressources sont auditées). La deuxième et la troisième exigences faciliteront la tâche de l'utilisateur autorisé, en lui montrant les informations et en lui fournissant certaines opérations de recherche par tri, de façon à lui faciliter la détection des évènements ou la combinaison d'évènements suspects.

2.1.7.5. La possibilité de sélection des évènements d'audit

La section enregistrement d'évènements d'audit de sécurité détermine les évènements à auditer lorsque le système fonctionne. Il est bien sûr évident que cette section tient compte des attributs de degré d'audit des utilisateurs et ressources du système pour déterminer la sélection des évènements.

Excepté pour un degré d'audit nul pour certaines ressources, cette exigence se retrouve toujours dans le cahier des charges car elle permet l'adaptation des évènements à auditer, selon l'expérience de l'entreprise (en effet, un système de sécurité possède toujours une faille), l'analyse des évènements à auditer en fonction des anciens et des nouveaux scénarios de violation par l'optimisation de la sélection des évènements.

- Le système doit pouvoir inclure ou exclure des évènements de l'ensemble $X_{1..N}$ des évènements à auditer en fonction d'un ensemble $Y_{1..N}$ d'attributs.

Exemple :
Le système doit pouvoir inclure ou exclure des évènements de l'ensemble des évènements à auditer en fonction *de l'identité de l'utilisateur.*

2.1.7.6. Enregistrement d'événements d'audit de sécurité

La section enregistrement d'évènements d'audit de sécurité permet au système de créer et de maintenir une trace d'audit sûre. En effet, pour pouvoir effectuer des revues d'audit performantes, il est conseillé de posséder un maximum d'informations pertinentes. Il faut aussi protéger les traces contre toutes défaillances ou modifications.

Excepté pour un degré d'audit nul pour certaines ressources, ces exigences se retrouvent dans tout cahier des charges car elles définissent les conditions d'enregistrement d'une trace d'audit, déterminantes pour la revue d'audit :

- Le système doit protéger les enregistrements d'audit contre toute suppression non autorisée.

- Le système doit pouvoir empêcher [(ou détecter)] des modifications effectuées sur les enregistrements d'audit.

  Exemple :
  Le système doit pouvoir *détecter* des modifications effectuées sur les enregistrements d'audit.

- Selon l'ensemble $X_{1..N}$ des conditions, le système doit garantir que la métrique Y des enregistrements d'audit sera maintenue.

  Exemple :
  En cas de dépassement de capacité du stockage, le système doit garantir que les traces d'enregistrements d'audit contiennent dans l'ordre : la date, le type d'événement, l'identification du responsable et puis les informations spécifiques au contexte d'utilisation.

- Le système doit effectuer une série $X_{1..N}$ d'actions dans le cas d'une défaillance possible dans le stockage des données d'audit, si la trace d'audit dépasse la limite Y.

  Exemple :
  Le système doit pouvoir effectuer *un backup des traces sur un support de secours* dans le cas d'une défaillance possible dans le stockage des données d'audit, si la trace d'audit dépasse la limite *de 350 mégas.*

- Quand la trace d'audit est pleine, le système doit entreprendre une ou plusieurs action(s) $X_{1..N}$.

  Exemple :
  Quand la trace d'audit est pleine, le système doit *ignorer les évènements à auditer.*

Pour que l'enregistrement s'effectue correctement, il faut que ces exigences garantissent l'intégrité et la disponibilité des fichiers de traces d'audit. Ainsi, les deux premières exigences concernent le stockage protégé des traces d'audit (accès limité à la lecture) et donc leurs protections contre les modifications et suppressions. La troisième exigence garantit la disponibilité des données d'audit, et donc leurs formats appropriés d'enregistrement. La quatrième et la cinquième exigence spécifient les actions à entreprendre en cas de perte éventuelle et pour la prévention de perte de données d'audit. L'ensemble de ces exigences répond bien à l'intégrité et à la disponibilité des fichiers.

2.2. Infrastructure et exigences techniques

2.2.1. Systèmes d'exploitation

Cette section doit être traitée au niveau de l'**ERP** car l'exigence se rapporte à l'entièreté du logiciel.

Le critère déterminant est **le degré de volonté de garder les systèmes d'exploitation existants**.

Si ce degré est important, alors cette exigence doit se retrouver dans le cahier des charges :

- Le système doit pouvoir fonctionner sur l'ensemble $X_{1..N}$ de systèmes d'exploitation.

  Exemple:
  Le système doit pouvoir fonctionner sur les systèmes d'exploitation *Windows 2000 et NT.*

2.2.2. Réutilisation des équipements informatiques existants (excepté protocoles réseaux)

2.2.2.1. Réutilisation Hardware

Ce point est important car il permet de déterminer la compatibilité entre l'équipement devant être réutilisé et le logiciel. En effet, il est conseillé d'indiquer l'existant hardware pour permettre aux fournisseurs d'évaluer l'équipement de l'entreprise.

Ce point doit être étudié au niveau de l'**ERP** car il se rapporte à l'ensemble du logiciel.

Les critères principaux du choix des équipements hardware à réutiliser est dans un premier temps **le degré de désir de réutiliser l'équipement hardware**. Ce degré dépend de facteurs budgétaires. En effet, il se peut que l'entreprise ne soit pas capable d'investir dans du nouveau matériel.

Si le degré de désir de réutilisation est important, alors cette exigence doit se retrouver dans le cahier des charges pour l'ensemble de l'équipement à réutiliser :

- Le système doit pouvoir réutiliser l'ensemble $X_{1..N}$ des équipements hardware actuels.

  Exemple :
  Le logiciel doit pouvoir réemployer *les postes clients (Pentium III 800, 256 Mo RAM).*

2.2.3. Réutilisation des réseaux

La partie réutilisation des réseaux spécifie les protocoles réseaux et les types de réseaux utilisés dans l'entreprise. Comme pour la réutilisation hardware, il s'agit ici de déterminer les compatibilités entre les composants du réseau existant avec le logiciel à acquérir.

Cette section doit être étudiée au niveau de l'**ERP** car les fonctionnalités d'un ERP n'ont pas de besoins spécifiques.

Le critère intervenant dans cette section est le **degré du désir de réutiliser les réseaux et protocoles existants**.

Si ce degré est important, alors l'exigence suivante doit se trouver dans le cahier des charges :

- Le système doit pouvoir fonctionner sur l'ensemble $X_{1..N}$ de type de réseaux et sur l'ensemble $Y_{1..N}$ des protocoles réseaux utilisés par l'entreprise.

  Exemple:
  Le logiciel doit pouvoir tourner *sur le réseau Ethernet*, avec comme protocoles *TCP IP, SMTP.*

2.2.4. Procédure d'installation et de test

Ensemble des exigences décrivant les contraintes d'installation du logiciel et les jeux de tests à fournir.

2.2.4.1. Contrainte d'installation

La partie contrainte d'installation spécifie qui est chargé de l'installation et qui est chargé du paramétrage logiciel.

Ce point est traité au niveau de l'**ERP** car l'installation et la configuration initiales concernent l'ensemble du logiciel.

Le critère intervenant dans ce point est **la complexité d'installation et de configuration**.

- Si l'installation et la configuration d'un ERP sont jugées très complexes (et c'est souvent le cas), ces exigences se retrouvent toujours dans le cahier des charges :

  - L'installation du logiciel doit être à la charge du fournisseur pour N postes clients.
  - La configuration du logiciel doit être à la charge du fournisseur pour N les postes clients.
  - L'installation du logiciel doit être à la charge du fournisseur pour N serveurs.
  - La configuration du logiciel doit être à la charge du fournisseur pour N serveurs.

avec dans ce cas, un N représentant la totalité du matériel (tous les postes clients, tous les serveurs).

- Si l'installation d'un ERP et sa configuration sont peu complexes, cette exigence se retrouve toujours, en plus des exigences du paragraphe précédent, dans le cahier des charges :

    - Le fournisseur doit spécifier les procédures d'installation et de configuration du logiciel pour les postes client et serveur.

Dans ce cas, le N est soit petit ou égal à 0.

2.2.4.2. Test

Ce point est traité au niveau des **fonctionnalités de l'ERP** car certaines fonctionnalités étant plus cruciales que d'autres, il est nécessaire de faire des jeux de tests plus importants pour celles-ci.

Le critère intervenant dans ce point est **le degré de criticalité des fonctionnalités**. Par degré critique, on entend les fonctionnalités qui sont essentielles à l'activité de l'entreprise.

L'exigence sur les jeux de tests à réaliser sera déterminée en fonction de ce degré :

- Si le degré de criticalité de certaines fonctionnalités est élevé, alors il est nécessaire de spécifier pour celles-ci des jeux de tests incluant un maximum de scénarios possibles. En effet, ces fonctionnalités étant critiques, il faut pouvoir s'assurer de la parfaite maîtrise de leur traitement. De plus, il est fortement conseillé d'effectuer des tests d'accès à cette fonctionnalité dans les cas critiques d'utilisation (selon les exigences sur les capacités de traitement).
  Par exemple, nous pouvons placer la planification des ordres de fabrication dans cette catégorie, car l'activité de l'entreprise dépend beaucoup de cette fonctionnalité.

    - Le fournisseur doit exécuter un ensemble $X_{1..N}$ de jeux de tests de degré Y pour l'ensemble $Z_{1..N}$ des fonctionnalités.

      Exemple pour la planification des ordres de fabrication :
      Le fournisseur doit exécuter *un ensemble de jeux de test très complet (y compris les conditions de fonctionnement extrêmes) pour la fonctionnalité de planification des ordres de fabrication.*

- Si le degré de criticalité de certaines fonctionnalités est faible, alors il est conseillé de spécifier pour celles-ci des jeux de tests incluant les scénarios classiques possibles (conditions de fonctionnement normal, etc.). En effet, ces fonctionnalités n'étant pas critiques, de simples jeux de tests suffisent.
  Nous pouvons donc y placer, par exemple, l'accès aux informations pertinentes des clients et la comptabilité analytique car l'activité de l'entreprise n'est en général pas mise en péril lors d'une défaillance de ces fonctionnalités.

  Exemple pour l'accès aux données pertinentes des clients:
  Le fournisseur doit exécuter *un ensemble de jeux de tests basiques pour la fonctionnalité « accès aux informations pertinentes des clients ».*

  Exemple pour la comptabilité analytique:
  Le fournisseur doit exécuter *un ensemble de jeux de tests basiques (conditions de fonctionnement normal) pour la fonctionnalité « comptabilité analytique ».*

2.3. Performances du système

2.3.1. La vitesse

Ensemble des exigences sur les contraintes temps d'affichage.

2.3.1.1. Les Interfaces

Cette exigence représente le temps nécessaire au passage d'une fenêtre à l'autre.

Ce point est traité au niveau de **l'ERP** car l'uniformité des interfaces est une caractéristique des logiciels de type ERP.

Le critère intervenant dans ce point est **le degré de confort d'utilisation souhaité** (conservation du flux de pensées des utilisateurs).

- Si le degré de confort souhaité est élevé, alors il est nécessaire de spécifier cette exigence avec un temps suffisamment court :

    - Le système doit pouvoir fournir pour toutes les interfaces un temps de passage d'un écran à un autre inférieur à X.

        Exemple:
        Toutes les fenêtres d'IHM devront s'afficher en moins de *N secondes* .

- Si le degré de confort est faible, alors il n'est pas nécessaire de spécifier cette exigence.

2.3.1.2. Temps de réponse

Ensemble des exigences relatives au temps de réponse pour l'obtention du résultat d'une demande (accès à des informations, recherche, calcul, etc.).

Ce point est traité au niveau des **fonctionnalités de l'ERP** car certaines fonctionnalités requièrent un temps de réponse inférieur à une certaine limite. Cette limite peut être importante afin d'éviter de bloquer tout processus opérationnel dépendant.

Le critère intervenant ici est le **désagrément provoqué par le retard d'un résultat**.
En effet, certaines tâches nécessitent d'avoir un temps maximum afin que celles-ci et leurs tâches dépendantes se déroulent normalement.

- Si le désagrément provoqué par le retard est important, alors il faut spécifier l'exigence type suivante :

    - Le système doit garantir un temps de réponse inférieur à X pour obtenir les résultats d'une fonctionnalité Y [ou d'un ensemble de fonctionnalité].

        Exemple pour l'accès aux informations pertinentes :
        Le logiciel doit garantir un temps de réponse inférieur à *N* pour obtenir les résultats de *l'accès aux informations pertinentes des clients.*

En effet, lors d'un contact avec un client (par téléphone, mail, etc.), il est très important que son interlocuteur puisse très rapidement avoir accès aux informations de ce client, afin de connaître l'historique des contacts avec ce dernier et pouvoir ainsi mieux le conseiller.

2.3.2. La précision

Les exigences de précision ont pour but de quantifier la précision désirée des résultats fournis par le logiciel afin d'éviter toute ambiguïté ou erreur.

Cette section est traitée au niveau des **fonctionnalités de l'ERP** car les types de données peuvent différer d'une fonctionnalité à une autre.

Le critère déterminant de cette section est **l'importance de la précision de certaines données.**

- Pour toutes les données dont la précision est jugée importante, il faut appliquer l'exigence type suivante :
    - X doit être précis à Y près.

Exemple pour la fonctionnalité de comptabilité analytique :
Les sommes monétaires doivent être précises à *deux décimales près.*

Tous les pourcentages seront précis à *trois décimales près.*

Exemple pour la planification des ordres de fabrication :
Toutes les quantités à produire doivent être précises à *une unité près.*

Exemple pour la fonctionnalité d'accès aux informations pertinentes des clients :
Les comptes clients doivent être précisés à *deux décimales près.*

2.3.3. Capacité de traitement et stockage de données

Ensemble des exigences visant à s'assurer que le logiciel sera effectivement capable de faire face à la charge de travail qui lui sera demandée.

2.3.3.1. Nombre d'utilisateurs

Ce point est traité au niveau de l'**ERP** car il s'agit du traitement pour un ensemble d'utilisateurs.

Le critère déterminant est le **nombre d'utilisateurs**.

- Si le nombre d'utilisateurs est faible, alors il n'est pas nécessaire de spécifier l'exigence type dans ce point car la plupart des ERP supporteront ce nombre d'utilisateurs.

- Si le nombre d'utilisateurs est élevé, alors il est nécessaire de spécifier l'exigence type dans ce point car il important que l'ERP puisse répondre à cette exigence.

    - Le système doit pouvoir supporter N utilisateurs simultanément [lorsque la période de temps est X].

        Exemple:
        Le logiciel doit pouvoir supporter *500 utilisateurs* simultanément.

2.3.3.2. Nombre de tâches

Ce point est traité au niveau des **fonctionnalités de l'ERP** car il s'agit du traitement pour un ensemble de tâches différentes ou identiques.

Le critère déterminant est **la quantité de tâches à réaliser simultanément**.

- Si la quantité de tâches à réaliser simultanément est faible, alors il n'est pas nécessaire de spécifier l'exigence type dans ce point car la plupart des ERP supporteront cette quantité.

- Si la quantité de tâches à réaliser simultanément est élevée, alors il est nécessaire de spécifier l'exigence type dans ce point car il important que l'ERP puisse répondre à cette exigence.

    - Le système doit être capable de traiter $X_{1..N}$ tâches simultanées [lorsque la période de temps est Y].

        Exemple pour la planification des ordres de fabrication :
        Le logiciel doit être capable de traiter *15 nouveaux ordres de fabrication simultanément.*

        Exemple pour la comptabilité analytique:
        Le logiciel doit être capable de traiter *une analyse de rentabilité pour 12 produits* simultanément *entre 9h00 et 10h00.*

        Exemple pour l'accès aux informations pertinentes des clients:
        Le logiciel doit être capable de traiter *120 accès aux informations des clients* simultanément *entre 9h00 et 10h00 et entre 16h00 et 17h30.*

2.3.3.3. Volume des données

Ce point est traité au niveau de l'**ERP** car il s'agit du traitement pour un volume de données.

Le critère déterminant est **le volume de données à traiter**.

- Si le volume de données à traiter est faible alors il n'est pas nécessaire de spécifier l'exigence type dans ce point car la plupart des ERP supporteront cette quantité.

- Si le volume de données à traiter est élevé alors il est nécessaire de spécifier l'exigence type dans ce point car il important que l'ERP puisse répondre à cette exigence.

- Le système est capable de traiter un volume X de données [lorsque la période de temps est Y].

  Exemple:

  Le logiciel sera capable de *traiter nos 600 clients actuels.*

  Le logiciel sera capable de *gérer nos 200 produits.*

2.3.4. Adaptation à une montée en charge

Ensemble des exigences visant à vérifier que le système sera capable de faire face à une augmentation des besoins. En effet, il faut pouvoir s'assurer que le logiciel sera capable de s'adapter à une augmentation du nombre d'utilisateurs, du nombre de tâches simultanées ou du volume de données. Rappelons que les exigences types de cette section sont identiques à celles de la section précédente à la différence près qu'il faut les conjuguer au futur.

Le critère déterminant pour cette section est **l'évolution dans le domaine d'activité de l'entreprise** (accroissement du nombre de clients, ajout de fonctionnalités supplémentaires, augmentation du volume des données, etc.).

- Si des évolutions sont prévues, alors il est nécessaire de spécifier les exigences de la section précédente suivie d'une date ou d'une période de temps future.

- S'il n'y a pas d'évolutions prévues, alors il n'est pas nécessaire de spécifier d'exigences dans cette section.

  Exemple pour le nombre d'utilisateurs:

  Le logiciel supportera simultanément *600 utilisateurs endéans une période de 4 mois.*

  Exemples pour le nombre de tâches :

  Exemple pour la comptabilité analytique:
  Le logiciel sera capable de traiter *une analyse de rentabilité pour 50 produits* simultanément *entre 9h00 et 10h00 dans six mois.*

  Exemple pour l'accès aux informations pertinentes des clients:
  Le logiciel sera capable de traiter *200 accès aux informations des clients* simultanément *entre 9h00 et 17h30 dans un an.*

  Exemple pour le volume des données:

  Le logiciel sera capable de *traiter 650 clients endéans une période de trois mois.*

2.4. Disponibilité

2.4.1. Tolérance aux pannes

Ce point est traité au niveau des **fonctionnalités de l'ERP** car la tolérance doit être différente par rapport à la criticalité des fonctionnalités.

Le critère déterminant est la **criticalité de la fonctionnalité** car plus la fonctionnalité est critique, plus une tolérance maximale aux pannes sera exigée pour celle-ci.

- Si la criticalité de la fonctionnalité est faible alors il n'est pas nécessaire de spécifier les exigences types dans ce point car elles ne sont pas essentielles à l'activité de l'entreprise.

- Si la criticalité de la fonctionnalité est élevée alors il est nécessaire de spécifier, dans ce point, les exigences types suivantes :

    - Le système doit garantir la disponibilité d'un ensemble $X_{1..N}$ de capacité lorsqu'un ou plusieurs éléments de l'ensemble $Y_{1..N}$ des défaillances survient.

        Exemple pour l'accès aux informations pertinentes des clients :
        Le logiciel doit garantir la disponibilité des *capacités de transfert et de sauvegarde des données en cours d'utilisation* soit lorsqu'une erreur *lors de l'exécution d'une fonctionnalité de l'accès aux informations pertinentes des clients* survient, soit lorsqu'une erreur *suite à une surcharge du système* survient.

    - L'ensemble $X_{1..N}$ des fonctionnalités du système doit être disponible pendant une période temps Y.

        Exemple pour la comptabilité analytique:
        La comptabilité analytique doit être disponible de *7h00 à 18h00 (heures d'ouverture du bureau).*

        Exemple pour la planification des ordres de fabrication :
        La planification des ordres de fabrication doit être disponible *24h/24h car nos chaînes de production ne s'arrêtent jamais.*

2.4.2. Priorité de service

La première exigence de ce point doit être traitée au niveau des **fonctionnalités de l'ERP** car elle a pour but de définir le niveau de priorité de ces fonctionnalités dans l'ERP. La deuxième exigence se traite au niveau de l'ERP car elle va obliger le système à vérifier que les priorités seront bien respectées.

Le critère déterminant est **l'uniformité des criticalités des fonctionnalités.**

- Si les fonctionnalités ont toutes le même niveau de criticalité (extrêmement rare), alors il est inutile de spécifier les exigences types de ce point.

- Si la criticalité des fonctionnalités n'est pas uniforme, alors il est nécessaire de spécifier, dans ce point, les exigences types suivantes :

- Le système doit attribuer une priorité X à un ensemble $Y_{1..N}$ de fonctionnalités du système.

  Exemple pour la comptabilité analytique:
  Le système doit attribuer une priorité *faible* à *la comptabilité analytique*.

  Exemple pour la planification des ordres de fabrication :
  Le système doit attribuer une priorité *forte* à *la planification des ordres de fabrication* (car cette fonctionnalité planifie la production des machines).

  Exemple pour l'accès aux informations pertinentes des clients :
  Le système doit attribuer une priorité *forte* à *l'accès aux informations pertinentes des clients* (car il faut un accès rapide pour visionner le profil du client).

- Le système doit garantir que chaque accès à un ensemble $X_{1..N}$ de ressources contrôlées doit être accordé sur base des priorités conférées à la fonctionnalité essayant d'y avoir accès.

  Exemple:
  Le logiciel doit garantir que chaque accès *à toutes ressources partageables* doit être accordé sur base des priorités conférées à la fonctionnalité tentant d'y avoir accès.

2.4.3. Allocation des ressources

Ensemble des exigences permettant d'éviter les dénis de service. En effet, il faut éviter qu'un processus prioritaire ne conserve trop longtemps les ressources hardware, pendant que d'autres, ayant une priorité égale ou inférieure, attendent sa terminaison.

Ce point est traité au niveau des **fonctionnalités de l'ERP** car l'allocation aux ressources dépend des priorités allouées aux fonctionnalités.

Le critère déterminant est **l'importance des désagréments provoqués par un déni de services**.

- Si le désagrément provoqué par la monopolisation d'une ressource est jugé faible, il n'est pas nécessaire de spécifier d'exigences dans ce point. Mais ce cas est en pratique très rare car il pourrait conduire à l'inutilisabilité pure et simple des fonctionnalités estimées moins critiques.

- Si le désagrément est considéré important, l'exigence ci-dessous doit se trouver dans le cahier des charges. Il est évident que seules les fonctionnalités de priorités élevées doivent être reprises dans ce point.

  - Le système doit limiter l'utilisation de l'ensemble de ressource $Y_{1..N}$ au quota X afin de s'assurer que la fonctionnalité Z ne les monopolise pas.

    Exemple pour la planification des ordres de fabrication :
    Le système doit limiter l'utilisation du *serveur de calcul* à *maximum N secondes* afin de s'assurer que la fonctionnalité de *planification des ordres de fabrication* ne le monopolise pas.

Exemple pour l'accès aux informations pertinentes des clients :
Le système doit limiter l'utilisation *de la base de données des clients* à *maximum N secondes* afin de s'assurer que la fonctionnalité *d'accès aux informations pertinentes des clients* ne la monopolise pas.

2.5. Fiabilité

Toute cette partie est traitée au niveau des **fonctionnalités l'ERP** car celles-ci ont des tolérances aux pannes admissibles différentes selon leurs importances dans l'activité de l'entreprise. Les réactions à prendre lorsque la défaillance survient peuvent également varier selon la fonctionnalité considérée.

Le critère déterminant pour l'ensemble de cette partie est donc **la criticalité de la fonctionnalité**.

2.5.1. Temps moyen entre deux pannes

- Si la criticalité des fonctionnalités est faible, alors il n'est pas nécessaire de spécifier les exigences types dans cette section car une panne ne mettra pas l'activité de l'entreprise en danger. On peut par exemple estimer que la criticalité de la comptabilité financière est faible et dans ce cas aucune exigence la concernant ne sera présente dans ce point.

- Si la criticalité des fonctionnalités est élevée alors il est nécessaire de spécifier, dans ce point, l'exigence type suivante :

    - L'ensemble $X_{1..N}$ des fonctionnalités du système doit avoir un MTBF [(*Mean Time Between Failure*)] de maximum Y.

        Exemple pour l'accès aux informations pertinentes des clients :
        L'accès aux informations pertinentes des clients doit avoir un MTBF de maximum N jours.

        Exemple pour la planification des ordres de fabrication :
        La planification des ordres de fabrication doit avoir un MTBF de maximum N mois car cette fonctionnalité est essentielle à l'activité de notre entreprise.

    La valeur du N varie évidemment selon la criticalité associée à l'exigence en question.

2.5.2. Temps d'action pour la réparation des défaillances

- Si la criticalité des fonctionnalités est faible, alors il n'est pas nécessaire de spécifier les exigences types dans cette section car une panne ne mettra pas l'activité de l'entreprise en danger.

- Si la criticalité des fonctionnalités est élevée, alors il est nécessaire de spécifier, dans cette section, l'exigence type suivante :

    - La réparation d'un ensemble $X_{1..N}$ de défaillances aura un MTTR [(*Mean Time To Repair*)] d'une période Y.

Exemple pour la planification des ordres de fabrication :
La réparation *d'une défaillance de la base de données des nouvelles commandes* doit avoir un MTTR de maximum N minutes car la fonctionnalité de planification des ordres de fabrication est essentielle à l'activité de notre entreprise.

Dans ce point, seules les défaillances sont considérées dans l'exigence type. Néanmoins, il faudra regarder quelles fonctionnalités cette défaillance affecte. En effet, une défaillance n'est pas grave en soi, mais ce sont ses répercutions sur les fonctionnalités critiques du système qui sont importantes.

2.5.3. Journal des problèmes

- Si la criticalité des fonctionnalités est faible, alors il n'est pas nécessaire de spécifier les exigences types dans cette section car de ces fonctions ne dépend pas la bonne activité de l'entreprise.

- Si la criticalité des fonctionnalités est élevée, alors il est nécessaire de spécifier, dans cette section, l'exigence type suivante :

  - Une traçabilité sera mise en place pour un ensemble $X_{1..N}$ d'évènements anormaux liés à des erreurs de programmation ou de traitement.

    Exemple pour la planification des ordres de fabrication :
    Une traçabilité sera mise en place pour *les quantités de production négatives ou nulles dans la planification des ordres de fabrication,* liés à des erreurs de programmation ou de traitement.

  Dans ce cas, il est également opportun de lier les évènements anormaux aux fonctionnalités auxquels ils se rapportent afin de refléter la criticalité de ces fonctionnalités.

  Rappelons que l'ensemble de ces évènements reste sélectionnable (voir point 2.1.7.5. : 'La possibilité de sélection des évènements d'audit').

2.6. Maintenance

2.6.1. Facilité de maintenance du produit

2.6.1.1. Personnes chargées de maintenance

Ensemble des exigences spécifiant les utilisateurs ou autres personnes responsables de la maintenance du système. Ce point est important car la maintenance d'un outil est souvent critique si celle-ci n'est pas réalisée par des personnes adéquates.

Ce point peut être traité au niveau de **l'ERP** car la maintenance peut concerner des modules pouvant contenir plusieurs fonctionnalités.

Cette exigence se retrouve dans tous les cahiers des charges d'ERP et **le degré de complexité de la maintenance** définira qui en sera chargé :

- Si la complexité de la maintenance est jugée élevée, alors il est nécessaire de spécifier cette exigence avec comme personne des spécialistes. En effet, seuls des spécialistes pourront garantir une maintenance sans encombre.

    - L'ensemble $X_{1..N}$ d'objet de maintenance devra être maintenu par $Y_{1..N}$.

        Exemple si pas de division en module:
        L'ensemble fonctionnalité du système devra être maintenu *par les développeurs de celui-ci.*

        Exemple si division en module :
        Le module comptabilité analytique devra être maintenu par *les développeurs de l'entreprise ayant suivis une formation.*

- Si la complexité de la maintenance est jugée faible, alors il est nécessaire de spécifier comme personne chargée de la maintenance, par exemple, des développeurs locaux.

    Exemple:
    L'ensemble des bases de données pourra être maintenu *par les responsables des bases de données de notre entreprise.*

2.6.1.2. Amélioration de la facilité de maintenance

Ensemble des exigences pour faciliter la maintenance. Cette partie concerne notamment des contraintes sur la façon dont le logiciel a été conçu, la documentation et les aides fournies pour la maintenance.

Ce point peut être traité au niveau de **l'ERP** car la maintenance concerne soit l'ERP dans son ensemble, soit les modules de celui-ci qui peuvent regrouper plusieurs fonctionnalités.

Le critère pour la première exigence de ce point est **le besoin de faire une maintenance sélective** car cela permet de faire la maintenance d'un module tout en continuant une exploitation normale des autres.

- Si ce besoin est jugé faible, alors il n'est pas nécessaire de spécifier dans ce point cette exigence type.

- Si ce besoin est jugé élevé, alors il est nécessaire de spécifier dans ce point l'exigence type suivante :

    - Le système sera divisé en ensemble de $X_{1..N}$ modules [et la maintenance d'un module pourra s'exécuter indépendamment des autres modules].

        Exemple :
        L'ERP sera divisé en des modules de *Gestion financière, Gestion des achats et stock, Gestion des relations clients, etc.* et la maintenance d'un module pourra s'exécuter indépendamment des autres modules.

Le critère pour la deuxième exigence de ce point est **le degré de complexité de la maintenance** car il définit quel acteur sera chargé de la maintenance.

- Si le degré de complexité est élevé, alors la maintenance sera confiée à une personne extérieure. Dans ce cas il ne sera pas nécessaire d'obtenir la documentation de maintenance. L'exigence type de ce point ne sera donc pas nécessaire.

- Si le degré de complexité est jugé faible, la maintenance sera effectuée en interne et elle nécessitera donc de la documentation. L'exigence type suivante devra donc se trouver dans le cahier des charges.

    - L'ensemble des documentations $X_{1..N}$ pour la maintenance doit être fourni et remis à jour.

        Exemple :
        La spécification du logiciel et la documentation d'aide à la maintenance seront fournis et mis à jour.

Il est à noter que généralement la maintenance d'un système ERP est très complexe et qu'elle sera donc souvent effectuée par des spécialistes extérieurs à l'entreprise.

2.6.2. Conditions spéciales de maintenance du produit

Ensemble des exigences relatives aux nouvelles versions.

2.6.2.1. Planning des fréquences et mises à jours

Ce point peut être traité au niveau de **l'ERP** car les nouvelles versions concernent l'ensemble de l'ERP.

Le critère pour ce point est **le degré d'évolution du logiciel.**
Le degré d'évolution du logiciel peut être évalué par la propension de l'environnement et de l'activité de l'entreprise à évoluer.

- Si l'importance d'évolution du logiciel est faible, alors il n'est pas nécessaire de spécifier les exigences types de ce point. Les mises à jour ne sont pas critiques et ne doivent pas être spécifiées dans le cahier des charges.

- Si l'importance d'évolution du logiciel est élevée, alors il est nécessaire de spécifier pour ce point, les exigences types suivantes :

    - Une mise à jour sera disponible tous les X sous la forme Y.

        Exemple :
        Une mise à jour sera disponible *tous les ans sous la forme de patch auto-exécutable fourni par le vendeur.*

    - Le fournisseur de logiciel doit fournir un ensemble de supports et de garanties $X_{1..N}$ pour l'installation des mises à jour.

        Exemple :
        Le logiciel doit fournir un *guide dactylographié et des garanties d'intégrité et de disponibilité* pour l'installation des mises à jour.

2.7. Apparence et perception : ergonomie et convivialité de la solution

Cette partie est traitée au niveau de l'**ERP** car l'apparence est globale à l'entièreté de l'ERP.

2.7.1. Interface graphique

Ensemble des exigences concernant l'esprit de l'interface (et non la conception de celle-ci en terme de programmation et en terme d'éléments affichés à l'écran). Ces exigences traitent du style d'écriture, des couleurs à utiliser, du degré d'interaction, etc.

Le critère intervenant dans cette section est le **degré d'importance accordé aux interfaces et aux interactions** avec celles-ci.

- Si ce degré est élevé, alors il est nécessaire de spécifier l'ensemble des exigences types suivantes :
  - L'interface doit utiliser un ensemble $X_{1..N}$ d'attributs graphiques.

    Exemple:
    Les interfaces seront *aux couleurs de la société.*
  - L'interaction entre l'utilisateur et les interfaces doit se faire selon une série $Y_{1..N}$ de dispositions.

    Exemple:
    L'interaction entre l'utilisateur et le système doit se faire *en langue française.*

    L'interaction entre l'utilisateur et le système doit se faire *en anglais, avec une interface de type Windows.*

  Ces exigences sont nécessaires lorsque le degré d'importance est fort car elles spécifient les contraintes graphiques selon l'environnement des utilisateurs : si le produit est destiné à une aide en ligne, si celui-ci s'adresse à une certaine catégorie d'utilisateurs, si le logiciel est utilisé dans des endroits très lumineux, etc. Il est donc nécessaire d'avoir un produit tenant compte de l'environnement.
- Si ce degré est faible, alors il est nécessaire de spécifier les exigences types, pour les langues supportées par le logiciel.

2.7.2. Le style du produit

Ensemble des exigences décrivant les caractéristiques d'accroche du produit, c'est-à-dire comment le produit sera perçu par son acheteur.

Le critère intervenant dans cette section est le **degré d'importance accordé aux interfaces et aux interactions** avec celles-ci.

- Si ce degré est élevé, alors il est nécessaire de spécifier l'exigence type suivante :
  - L'interface doit utiliser un ensemble $X_{1..N}$ de styles caractéristiques.

Exemple :
L'interface utilisera un style S (par exemple contemporain).

- Si ce degré est faible, alors il n'est pas nécessaire de spécifier l'exigence type.

2.7.3. Facilité d'utilisation et aide

Ensemble des exigences décrivant la facilité d'utilisation du logiciel ainsi que la facilité d'apprentissage. Cette facilité est importante pour rendre l'utilisation de l'ERP aisée pour les utilisateurs.

Le critère pour cette section est **la compétence des futurs utilisateurs par rapport aux tâches qu'ils devront effectuer**.

- Si la compétence des utilisateurs est élevée, alors il n'est pas nécessaire de spécifier les exigences de cette section car ces utilisateurs seront aptes à employer l'ERP sans aide.

- Si la compétence des utilisateurs est moyenne ou faible, alors il est nécessaire de leur permettre de s'adapter à l'ERP et de les aider lors de l'utilisation du logiciel. Pour ce faire, les exigences types suivantes sont nécessaires :

  - Le système doit pouvoir être utilisé pour une certaine catégorie X d'utilisateurs ayant suivi au maximum des types $Y_{..N}$ d'adaptation pour l'utilisation.

    Exemple :
    Le système doit pouvoir être utilisé par des *utilisateurs de compétence moyenne* ayant suivi au maximum *une formation d'une semaine.*

  - Le système doit guider l'utilisateur, avec un ensemble X de mécanismes, afin d'éviter qu'il ne commette des erreurs.

    Exemple :
    Le système doit guider l'utilisateur, avec *des bulles d'aides contextuelles*, pour éviter qu'il ne fasse des erreurs.

  - Le système doit être accompagné d'un ensemble $X_{1..N}$ de supports, accessibles à des temps Y, pour l'aide à l'utilisateur.

    Exemple :
    Le logiciel doit être accompagné *d'une hot-line pouvant être joignable de 6H00 à 22H00 afin d'aider* l'utilisateur.

    Le logiciel doit être fourni *avec un tutorial.*

2.8. Interfaçage de données d'un logiciel à l'autre

Ensemble des exigences spécifiant l'ensemble des applications et leurs données avec lesquelles l'ERP devra s'interfacer. Il s'agit donc ici de déterminer si certaines de ses applications peuvent s'interfacer avec le logiciel ERP. Par définition, un ERP concerne l'intégration complète des fonctions de gestion de l'entreprise.
Toutefois, il peut s'avérer nécessaire d'interfacer les données d'autres logiciels (par exemple, avec un logiciel pour la gestion automatique des stocks via des scanners laser) pour bénéficier de leurs informations et les intégrer dans l'ERP.

Cette section doit être traitée au niveau des **fonctionnalités de l'ERP** car chacune de ces fonctionnalités peut avoir des besoins d'interfaçage très différents. Il est donc plus aisé de répertorier ces logiciels par fonctionnalité.

Le critère pour cette partie est **le besoin d'interfacer des logiciels dont les fonctionnalités ne font pas partie d'un ERP**.

- Si le besoin est inexistant, alors il n'est pas nécessaire de spécifier l'exigence de cette section.

- Si le besoin existe, alors il est nécessaire de spécifier les exigences types suivantes :

    - Le système doit s'interfacer avec [un ensemble $W_{1..N}$ de données d'] un ensemble $X_{1..N}$ d'applications [à une fréquence Y] [via un médium Z utilisé pour l'interfaçage].

        Exemple pour la comptabilité analytique :
        L'ERP doit pouvoir *s'interfacer aux données (dont le volume est cinq mégas et le format est un format propriétaire) de comptabilité d'une ancienne version de ce logiciel tous les jours, via le réseau Ethernet de la société.*

        Exemple pour la gestion du planning de fabrication:
        Le logiciel doit pouvoir s'interfacer avec *un logiciel prévisionnel complexe spécialisé dans le domaine d'activité de l'entreprise.*

    - Le système doit fournir des standards $X_{1..N}$ d'échanges de données.

        Exemple :
        Le logiciel doit supporter les standards OLE, XML d'échanges de données.

La première exigence précise les données et, si possible, la fréquence et le médium à utiliser pour le transfert. La seconde exigence indique s'il existe des standards compatibles pour l'échange de données. En effet, le client qui utilise certains standards de données, exigera de les utiliser.

2.9. Intégration dans la nouvelle base de données.

Ensemble des exigences traitant de l'intégration de données dans la nouvelle base de données.

Cette section doit être traitée au niveau **des fonctionnalités de l'ERP** car l'intégration des données diffère selon la compatibilité des formats de données actuelles et futures.

Le critère pour l'exigence suivante est **l'existence de données à intégrer**.

- S'il n'existe pas de données à intégrer, alors il n'est pas nécessaire de spécifier les exigences types de cette section.

- S'il existe un ensemble de données à intégrer, alors il est nécessaire de spécifier l'exigence type suivante :

  - La base de données doit pouvoir intégrer l'ensemble de données $X_{1..N}$.

    Exemple pour la comptabilité analytique :
    La base de données de l'ERP doit pouvoir intégrer l'ensemble des données de la comptabilité analytique actuelles.

    Exemple pour l'accès aux informations pertinentes des clients :
    La base de données doit pouvoir intégrer les informations actuelles des clients ainsi que l'historique de nos précédents contacts.

Le critère pour l'exigence suivante est **la quantité d'informations à transférer**.

- Si la quantité est faible, alors il n'est pas nécessaire de spécifier les exigences types de cette section.

- Si la quantité est importante, alors il est nécessaire de spécifier l'exigence type suivante :

  - X est chargé de la transposition de l'ensemble $Y_{1..N}$ des données actuelles [en utilisant pour ce transfert une méthode Z de transposition].

    Exemple pour la comptabilité analytique:
    Le fournisseur est chargé de la transposition de l'ensemble des données clients en utilisant pour ce transfert *des requêtes SQL*.

    Exemple pour l'accès aux informations pertinentes des clients:
    Le fournisseur est chargé de la transposition de l'ensemble des données clients.

La méthode de transposition dépend du nombre de données à transférer et de la compatibilité de ces données (par exemple, format réel vers entier). Plus la quantité à transférer est grande, plus il sera nécessaire de se diriger vers une solution automatique (requêtes SQL par exemple). Plus la compatibilité est faible, plus il sera difficile d'utiliser une méthode automatique.

La première exigence est relative à l'intégration et la seconde à la méthode utilisée pour la transposition.

2.10. Exigences culturelles et politiques

Cette partie contient l'ensemble des exigences spécifiques aux facteurs sociologiques et de politiques internes qui affectent l'acceptabilité du produit. Une entreprise n'étant pas une autre, certaines contraintes peuvent s'ajouter au système à acquérir.
Il est donc nécessaire de spécifier dans cette section, les différentes contraintes liées à la culture et à la politique de l'entreprise.

Cette section doit être traitée au niveau de **l'ERP** car toutes les exigences suivantes ont une portée sur l'entièreté de l'ERP.

Le critère pour cette section est **l'existence de contraintes culturelles et politiques**.

- S'il existe des contraintes culturelles et politiques, alors il est nécessaire de les spécifier au moyen des exigences types suivantes :

    - Le système doit être en accord avec un ensemble $X_{1..N}$ d'exigences culturelles.

        Exemple :
        Le système doit supporter les *métriques européennes (kilo, mètre, etc.)*.

    - Le système doit être en accord avec un ensemble $X_{1..N}$ de données politiques.

        Exemple :
        Le système doit être en accord avec les *exigences politiques internes à l'entreprise qui sont* ....(à spécifier selon le cas)

2.11. Contraintes légales et normes

Cette section doit contenir toutes les exigences concernant les normes ou règlements auxquels le logiciel doit répondre.

2.11.1. Exigences légales

Cette section doit être traitée au niveau de **l'ERP** car c'est le logiciel dans son entièreté qui doit être en conformité avec la loi.

Le critère pour cette section est **l'existence de contraintes légales**.

- S'il existe des contraintes légales, alors il est nécessaire de les spécifier au moyen de l'exigence type suivante :

    - Le système doit être en conformité avec la loi X.

        Exemple :
        Le système doit être en conformité avec loi du 2 août 2002 dite "sur la vie privée".

2.11.2. Normes

Cette section doit être traitée au niveau de **l'ERP** car toutes les exigences suivantes ont une portée sur l'entièreté de l'ERP.

Le critère pour cette section est **l'existence de contraintes normatives.**

- S'il existe des normes contraignantes, alors il est nécessaire de les spécifier au moyen de l'exigence type suivante :

  - Le système doit être conforme à un ensemble $X_{1..N}$ de standards.

    Exemple :
    Le système doit être en conformité avec les certifications Iso 15408, Comptabilité Euro certifiée, etc.

Conclusion

Ce chapitre a permis de mettre en avant l'utilisation de l'état de l'art effectué au chapitre 1. En effet, nous avons défini ici certains critères permettant de spécifier si une exigence type (ou un groupe d'exigences) devait se trouver ou non dans le cahier des charges.

Nous avons essayer de poser le moins possible de jugements de valeurs afin de garder un certain niveau de généricité (Nous n'avons pas par exemple jugé que tous les ERP ont un nombre d'utilisateurs élevé même si cela est souvent le cas). Nous avons néanmoins émis quelques jugements mais cela juste à des fins d'exemples. (Nous avons parfois jugé de la criticalité des fonctionnalités).

L'exemple qui a été développé n'est évidemment pas exhaustif mais est néanmoins suffisant pour percevoir l'apport de la méthodologie appliquée. En effet, lors de la création d'un cahier des charges, il suffit d'estimer les critères définis (nombre d'utilisateurs, criticalités des données, etc.) pour être capable d'en déduire facilement les exigences à spécifier. La méthodologie appliquée permet donc un gain de temps considérable et une certaine systématisation de la production des exigences non fonctionnelles.

Il est évident que la méthodologie appliquée ici peut aussi convenir à d'autre domaine que celui donné en exemple.

Chapitre 3 : Partie pratique, le logiciel OPAL

Introduction

La création de cahiers des charges et de l'animation de l'appel d'offres est très complexe dans le domaine informatique. C'est pour cette raison que nombres de logiciels ont été créés afin d'aider le consultant dans sa démarche.

Le Centre de Recherche Henri Tudor (Luxembourg), conscient de cette complexité, a mis sur pied le projet SPINOV (Software Process Improvement and inNOVation). Ce projet a pour objectif de développer et maintenir des compétences dans le domaine de l'ingénierie des exigences, discipline sous-tendant la gestion des cahiers des charges.

C'est dans SPINOV que le projet OPAL est né. Son but premier est la création d'un logiciel assez complet capable d'aider le consultant dans la création première de son cahier des charges et de lui fournir des outils suffisants pour animer l'appel d'offres.

OPAL n'a évidemment pas la prétention de remplacer d'autres logiciels beaucoup plus spécialisés tels que :

Analyst Pro [10]

Outil très complet édité par Goda Software dont les fonctionnalités principales sont :

- **Requirements Management** : Permet de créer, classifier et importer des exigences.
- **Traceability Analysis** : Aide à étudier l'impact de changements sur les exigences. Il est en outre possible d'étudier l'impact des changements selon plusieurs perspectives.
- **Gestion des versions** des exigences (avec la possibilité de consulter un historique des versions).
- **Génération de rapports** : Exigences, rapport de changements, etc.
- **Test Cases management** : Permet de créer des jeux de test afin de vérifier la qualité du logiciel. Ces jeux de test peuvent être de type :

    1. Requirements Testing
    2. White Box Testing
    3. Black Box Testing
    4. Regression Testing and
    5. Code Walk-Through.

- **Workflow management** : Permet une communication efficace entre tous les acteurs du projet.

[10] http://www.analysttool.com/products.html

Requisite Pro [11]

Ce logiciel, édité par rational[12] (IBM), est un outil de gestion des exigences largement interfacé avec Microsoft Word. Ses principales fonctionnalités sont les suivantes :

- **Requirements management :** Permet de créer, classifier et importer des exigences.
- **Impact Analysis and Change Management :** Permet l'évaluation de l'impact des changements effectués aux exigences.
- **Flexible Reporting :** Permet de générer différents documents comme par exemple des rapports de traçabilité, etc.

  Requisite Pro peut être facilement interfacé avec les nombreux autres logiciels de Rationnal comme par exemple IBM Rational Suite (outil permettant entre autres de gérer la progression de projet).

Le programme OPAL se trouve donc en amont de ces logiciels très évolués et ne désire en aucun cas s'y substituer. En effet, dans OPAL, l'accent est plutôt mis sur la réutilisation et ce, par un système évolué de capitalisation/réutilisation des projets menés.

Dans ce chapitre, nous vous présentons un aperçu de l'outil OPAL, de sa structure et de ses fonctionnalités. Ce logiciel se trouve sur le CDRom fourni avec ce mémoire. Une aide à l'installation ainsi qu'un rapide exemple d'utilisation se trouve en annexe 3.

3.1 Fonctionnalités OPAL

Le but de ce point est de décrire les fonctionnalités principales du logiciel. OPAL a été partagé en 2 modules autonomes mais complémentaires afin de diviser ses fonctionnalités en :

- Les fonctionnalités générales qui permettent la création et la construction d'un cahier des charges ainsi que l'animation de l'appel d'offres.
  → OPAL CONSULTING
- Les fonctionnalités plus délicates et ayant un impact sur l'ensemble des projets OPAL.
  → OPAL ADMIN

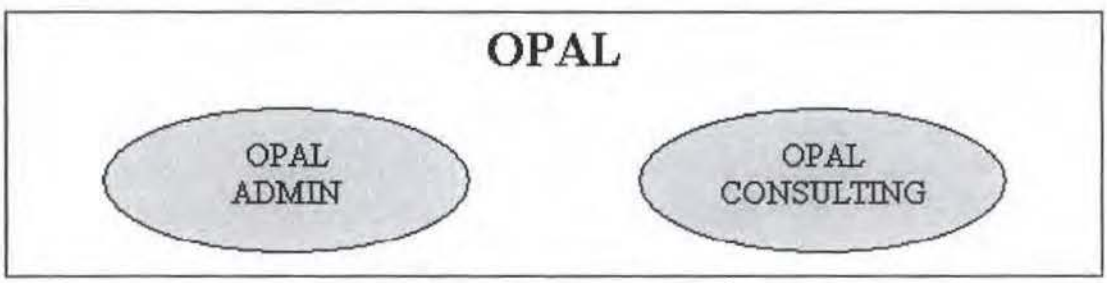


Figure 2 : Division des fonctionnalités de OPAL

[11] http://www.rational.com/products/reqpro/
[12] http://www.rational.com/

De plus, afin de protéger l'accès aux fonctionnalités sensibles de OPAL, 3 classes d'utilisateurs ont été définies :

1/ L'utilisateur simple

Un consultant OPAL n'a accès qu'au module 'Consulting' du programme. Il a donc la possibilité de créer des cahiers de charges ainsi que de sauvegarder ses préférences de projets en tant que modèle de préférences qu'il peut réutiliser par la suite et partager avec d'autres consultants.

2/ L'administrateur

L'administrateur possède tous les droits d'un consultant. En outre, il a accès au module 'administrateur' de OPAL. Il lui est donc permis de créer de nouveaux comptes consultant, etc.

3/ Le super administrateur

Il possède tous les droits de l'administrateur mais a également la possibilité de créer de nouveaux administrateurs.

Avant de décrire les fonctionnalités principales du logiciel, nous présentons ci-dessous une liste des concepts clés et du vocabulaire qui sera utilisé lors de la description des fonctionnalités.

Vocabulaire et concepts clés

- **Degré de partage d'un projet** : Le degré de partage d'un projet reflète les droits d'accès des autres utilisateurs à ce projet. Ces droits peuvent être de quatre types :

    1. Aucun : Seul l'utilisateur ayant créé le projet a le droit de l'ouvrir et de le réutiliser afin d'importer des exigences dans un autre projet.

    2. Partager comme modèle : Seul l'utilisateur ayant créé ce projet a le droit de l'ouvrir et de le modifier. Néanmoins, les autres utilisateurs peuvent importer des exigences provenant de ce projet.

    3. Partager : Tous les utilisateurs ont le droit d'ouvrir et de modifier le projet. Ils ont également tous le droit d'importer des exigences à partir de ce projet.

    4. Template : Tous les utilisateurs ont le droit d'ouvrir et de modifier le projet mais ils ne pourront en aucun cas l'enregistrer tel quel. Ils devront au préalable changer le nom sous lequel ils désirent sauvegarder leur projet afin de ne pas modifier un template. (non encore implémenté) .

- **Business domain** : Catégorie précise de logiciel (par exemple CRM, ERP, etc.) qui est utilisée afin de catégoriser les projets et les rendre ainsi plus facilement accessibles.

- **Système de pondération** : Ensemble des pondérations possibles d'une exigence (par exemple, accessoire, importante, très importante). Ce système va donc permettre de gérer l'importance relative des exigences entre elles.

- **Système de notation** : Ce système permet de quantifier la réponse d'un fournisseur par rapport à une exigence. Par exemple, exigence totalement satisfaite, partiellement satisfaite ou non satisfaite.

- **Glossaire** : Ensemble de termes utilisés dans le cahier des charges associés à leurs définitions.

- **Structures de description d'une exigence** : Structure via laquelle une exigence va être décrite. Dans OPAL, il est possible de définir la structure de description utilisée pour chaque partie du cahier des charges (Présentation, exigences fonctionnelles, exigences non fonctionnelles, critères d'appel d'offres).

- **Modèle de préférences** : Ensemble rassemblant un système de pondération, un système de notation, un glossaire ainsi qu'un ensemble de structures de description (une pour chaque partie du cahier des charges).

- **Terminology Set** : Ensemble reprenant la 'traduction' d'un certain nombre de keywords afin d'être affichée dans l'IHM d'OPAL. Le keyword EXF peut par exemple être associé au terme 'exigences fonctionnelles' afin que ce terme soit affiché dans un onglet.

- **Activité métier** : Activité liée à un type de métier. Les différentes exigences du cahier des charges peuvent être liées à une ou plusieurs activités métiers afin d'assurer un certain niveau de traçabilité.

- **Catégorie** : Catégorie-packaging d'éléments du cahier des charges (par exemple, IHM). La catégorisation d'une exigence permet d'assurer un certain niveau de traçabilité des exigences.

- **Goal** : But ou sous-but d'un projet. Les goals sont utilisés pour classifier les exigences d'un cahier des charges afin d'assurer un certain niveau de traçabilité des exigences. Un goal peut être par exemple 'augmenter la compétitivité de l'entreprise'.

- **Concepts Avancés** : Ensemble reprenant les activités métiers, les catégories et les goals d'un projet.

3.1.1 OPAL Consulting

Le module Consulting de OPAL est un module permettant de réaliser le but premier de OPAL, à savoir la création de cahiers des charges et l'animation de l'appel d'offres. Ce module est accessible à tous les utilisateurs OPAL (utilisateurs simples, administrateurs et super administrateurs).

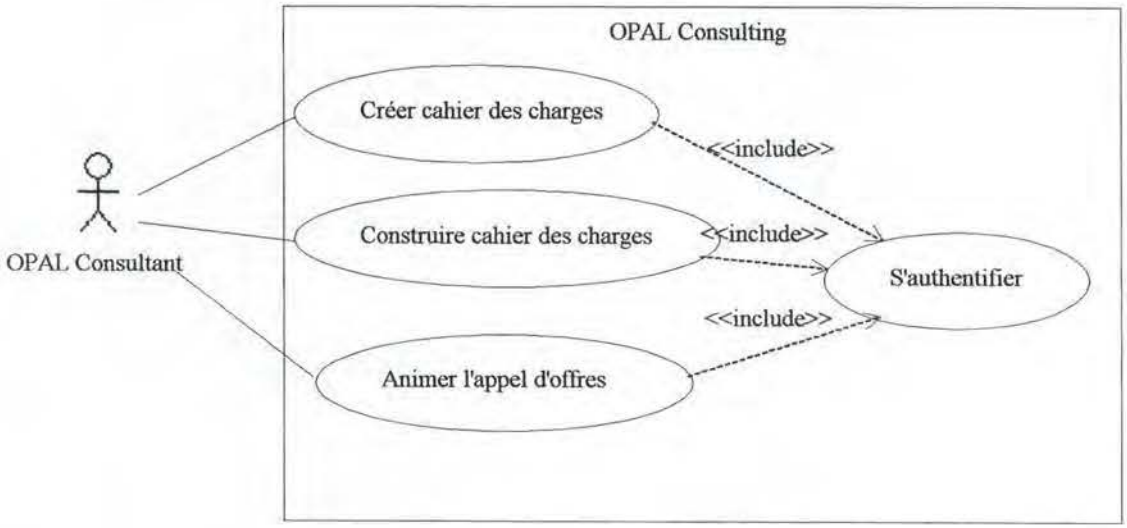


Figure 3 : Use case général OPAL Consulting

3.1.1.1 S'authentifier

Cette fonctionnalité permet d'authentifier (login/password) tout utilisateur de OPAL Consulting. Cette authentification est obligatoire et a lieu à chaque démarrage de l'application. L'utilisateur a également la possibilité de changer les informations le concernant (Nom, Prénom, etc.) dans la base de données administrateur. Ces informations sont en outre reprises dans le volet consultant des informations générales d'un projet.

3.1.1.2 Créer un cahier des charges

Lors de la création d'un nouveau cahier des charges, plusieurs informations doivent être fournies afin de caractériser le projet et en définir les droits de partage. Comme indiqué dans la capture d'écran ci-dessous, les informations à fournir sont les suivantes :

- Le business domain du projet, par exemple ERP, CRM, etc. Cette typologie de logiciel est gérée par l'administrateur et a pour but de faciliter le choix des modèles utilisés lors de la réutilisation des contraintes dans un projet courant.
- Le nom du projet.
- Une description facultative du projet. Cette description est le premier contact que tout utilisateur ouvrant un projet aura avec lui.
- Le modèle de préférences souhaité.
- Le degré de partage du projet.

La figure 4 montre l'IHM permettant de créer un nouveau projet dans OPAL Consulting.

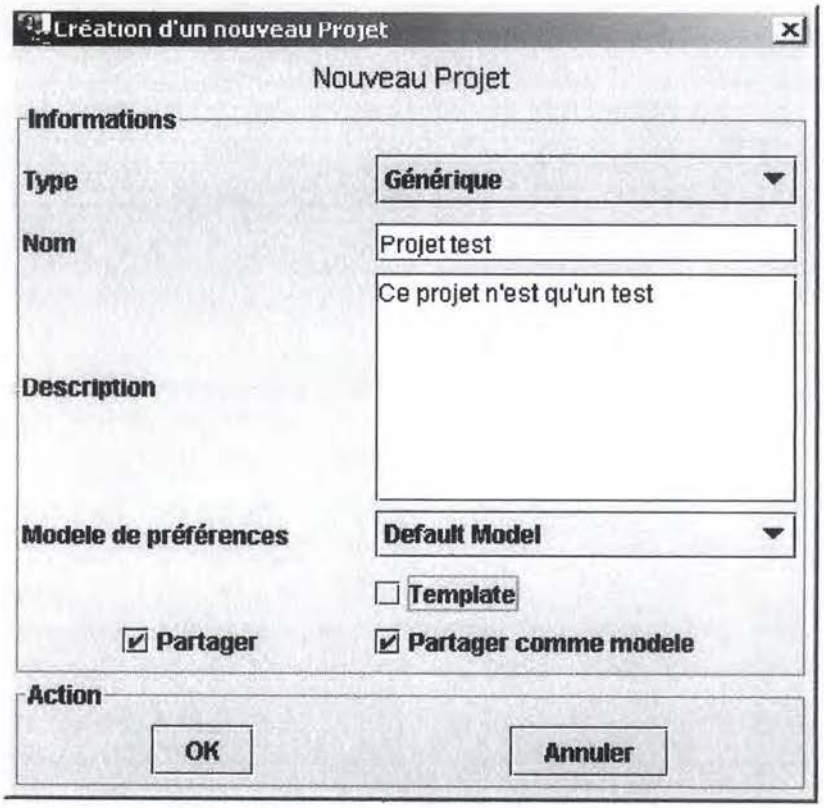


Figure 4 : IHM Création nouveau projet

3.1.1.3 Construire cahier des charges

La construction du cahier des charges a été divisée en 3 graphiques afin de rester lisible (Figure 5, 9, 11).

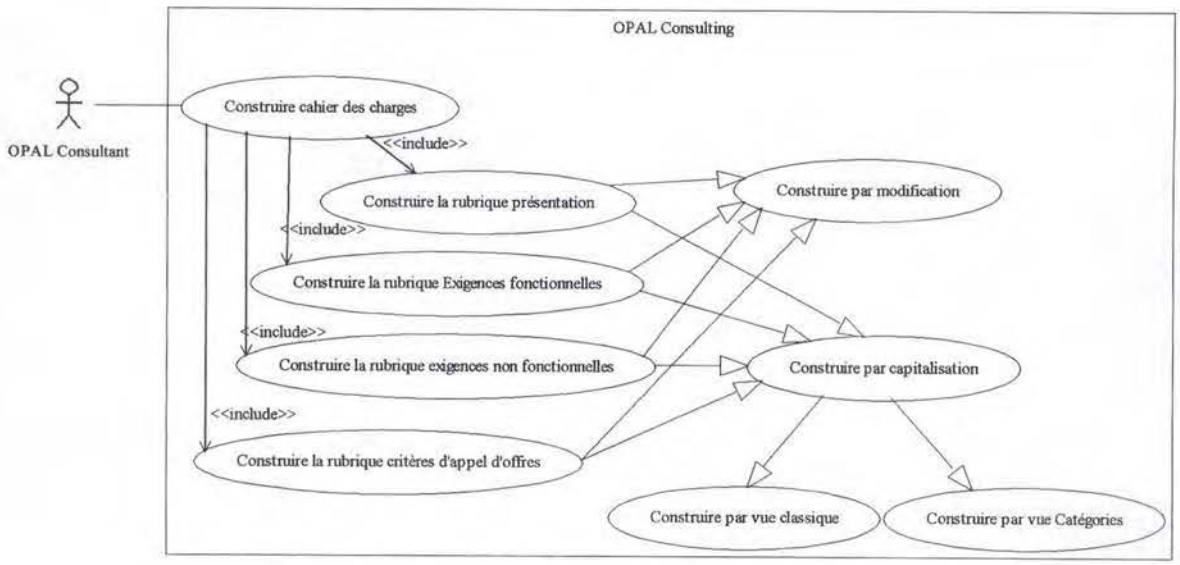


Figure 5 : Use case diagram "Construire cahier des charges" 1

- Construire la rubrique présentation

Permet de construire/modifier la partie présentation du projet courant.

- Construire la rubrique Exigences fonctionnelles

Permet de construire/modifier la rubrique des exigences fonctionnelles.

- Construire la rubrique Exigences non fonctionnelles

Permet de construire/modifier la rubrique exigences non fonctionnelles.

- Construire la rubrique Critères d'appel d'offres

Permet de construire/modifier la rubrique des critères d'appel d'offres.

- Construire par vue 'classique'

Lors de la construction par capitalisation d'une partie, l'utilisateur peut choisir la vue classique (un élément fils étant relié à son élément père). La structure affichée à l'écran est donc un arbre classique.

- Construire par vue 'catégories'

Il est également possible d'afficher le projet courant ou le modèle de capitalisation en vue 'catégories'. Dans cette vue, tous les éléments sont directement reliés à la catégorie qui leur a été assignée dans la matrice de catégorisation. Les éléments non catégorisés sont reliés aux roots de l'arbre.

- Construire par capitalisation

L'utilisateur peut intégrer au projet en cours des éléments d'anciens projets pour autant que les droits de partage de ceux-ci soient suffisants ('Partager comme modèle'). Un diagramme d'activités représentant la construction par capitalisation se trouve ci-dessous (Figure 6).

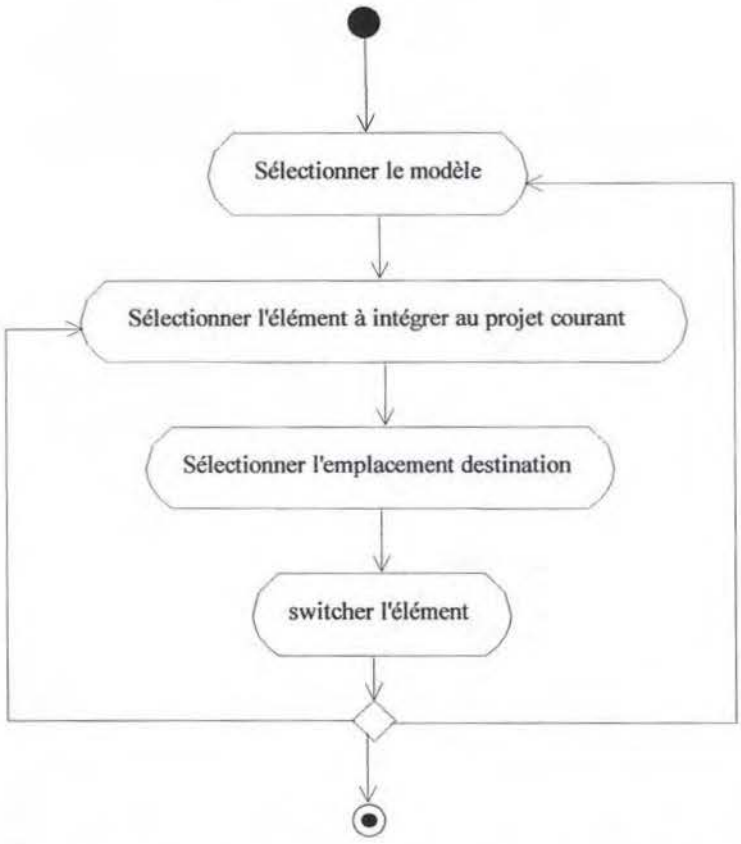


Figure 6 : Diagramme d'activités 'Construire par capitalisation'

Lors de la capitalisation, le contenu de l'élément à copier sera également transféré. Néanmoins, si les structures de descriptions diffèrent, le contenu sera automatiquement transformé en simple texte et inséré dans l'exigence nouvellement copiée (dans un champ prévu à cet effet).

- Construire par modification

L'utilisateur a la possibilité de modifier la structure courante via des fonctionnalités d'ajout/suppression/déplacement. La figure 7 présente un diagramme d'activités indiquant les actions possibles.

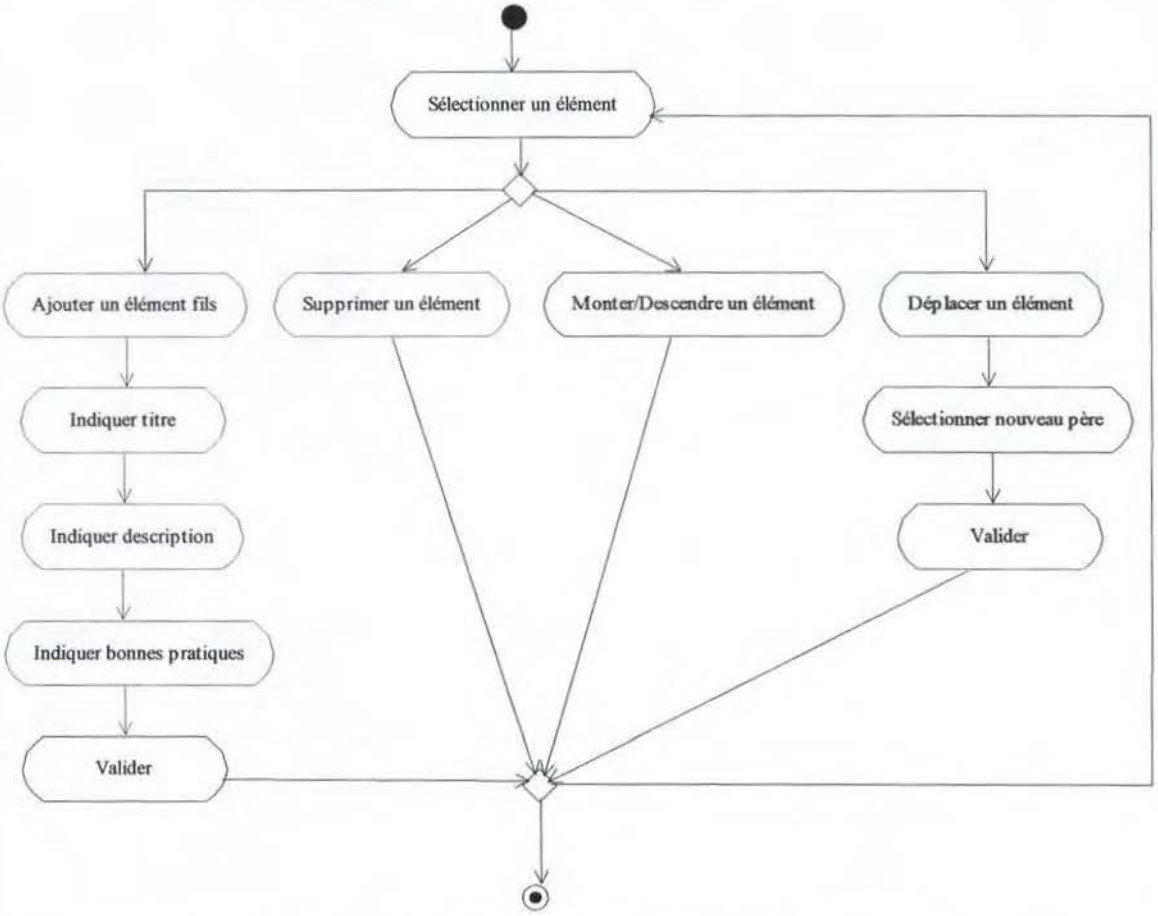


Figure 7 : Diagramme d'activités 'Construire par modification'

La Figure 8 représente l'interface permettant la construction du cahier des charges ainsi que des critères d'appel d'offres.

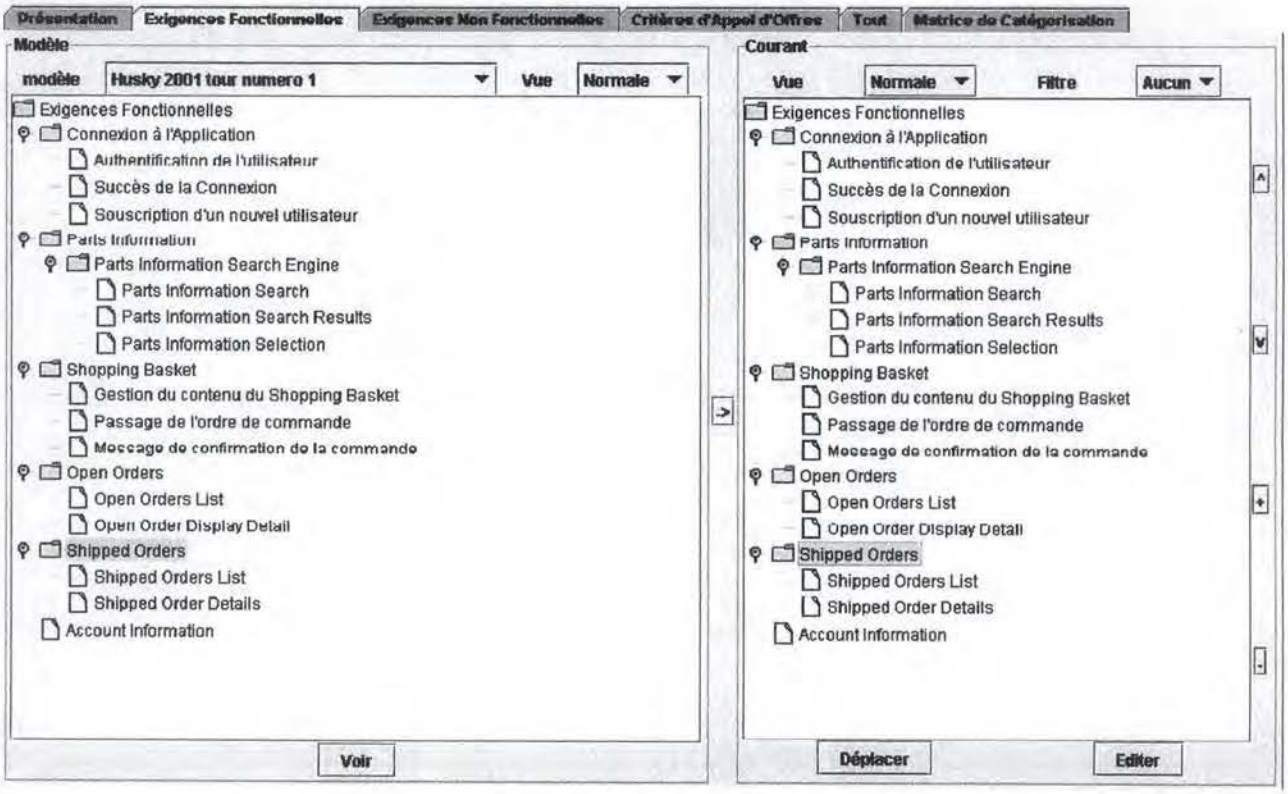


Figure 8 : IHM de construction CDC/AO

La création de cahier des charges intègre aussi la pondération des exigences et la gestion de la traçabilité comme montré par la Figure 9 :

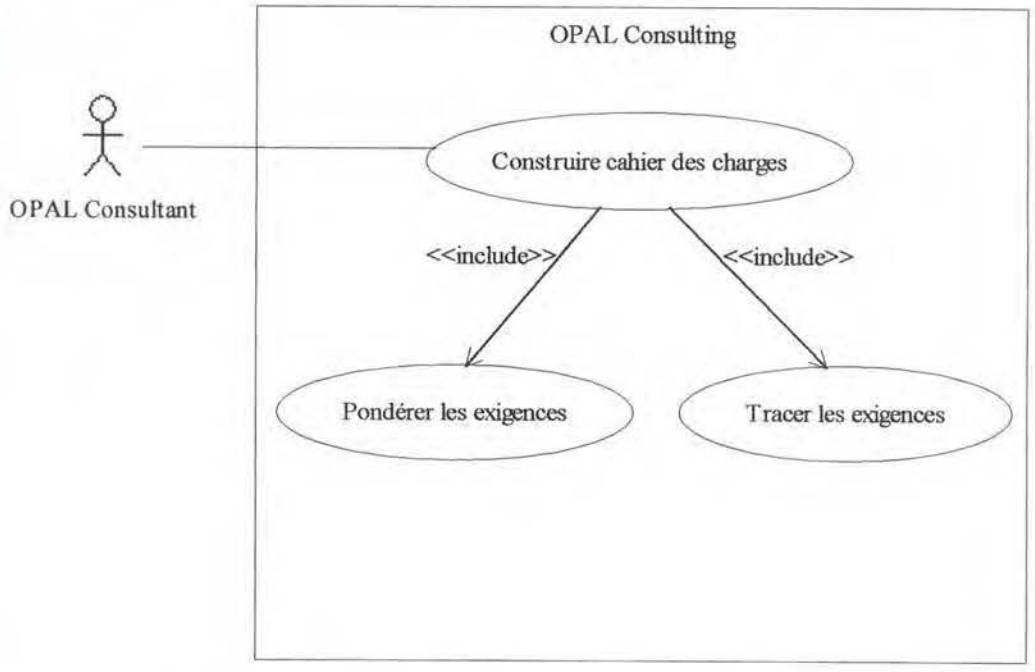


Figure 9 : Use case diagram "Construire cahier des charges" 2

- Pondérer les exigences

Cette fonctionnalité permet d'associer une pondération (c'est-à-dire un élément du système de pondération) à un élément des exigences fonctionnelles, exigences non fonctionnelles ou critères d'appel d'offres.
Cette pondération représente l'importance relative d'un élément par rapport aux autres éléments du même type.

- Tracer les exigences

L'utilisateur peut associer les exigences fonctionnelles, exigences non fonctionnelles ou critères d'appel d'offres à un ou plusieurs éléments des concepts avancés. A l'heure actuelle, seules les catégories ont été implémentées.
L'importance relative de l'élément dans une catégorie se traduit par une pondération entre cette catégorie et l'élément en question. La Figure 10 représente l'écran permettant cette pondération :

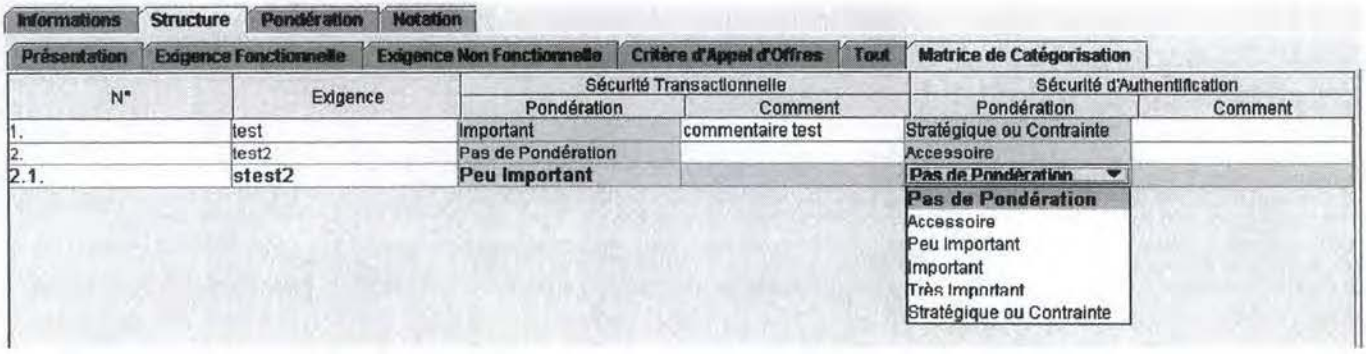


Figure 10 : IHM Matrice de catégorisation

La Figure 11 présente les dernières fonctionnalités comprises dans la création du cahier des charges :

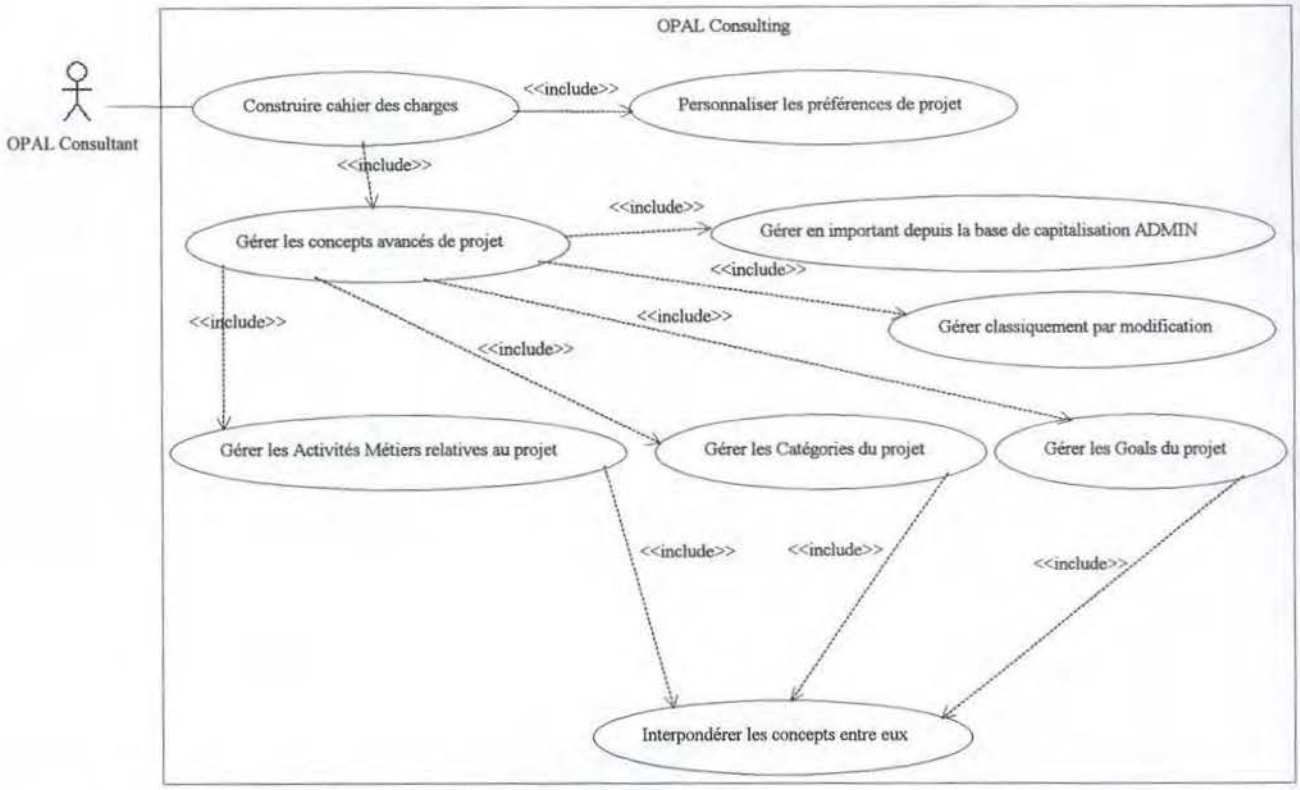


Figure 11 : Use case diagram "Construire cahier des charges" 3

- Gérer les concepts avancés de projet

La gestion des concepts avancés d'un projet (activités métiers, catégories, goals) permet l'obtention d'un ensemble d'activités métiers, catégories, goals dont les relations avec les différentes exigences pourront être tracées.

- Gérer les activités métiers relatives au projet

Permet l'obtention d'un ensemble d'activités métiers liées au projet. (Non encore implémenté).

- Gérer les catégories du projet

Permet l'obtention d'un ensemble de catégories liées au projet.

- Gérer les goals du projet

Permet l'obtention d'un ensemble de goals liés au projet. (Non encore implémenté).

- Interpondérer les concepts entre eux

Dans OPAL il est possible de préciser une pondération entre chaque élément des catégories, goals ou activités métiers. Cette pondération pourra être utilisée lors de l'évaluation des différents fournisseurs. Ce point est très semblable à l'interpondération des grandes parties du cahier des charges mais n'a pas encore été implémenté.

- Gérer en important depuis la base de capitalisation ADMIN

Le consultant OPAL peut importer des catégories, goals, activités métiers directement depuis la base de données administrateur. Il lui suffit de sélectionner un type de business domain et les éléments qu'il désire importer.

- Gérer classiquement par modification

Cette fonctionnalité permet au consultant de définir lui-même l'ensemble des éléments d'un concept avancé. La Figure 12 présente l'écran permettant de réaliser cet ensemble via des fonctionnalités d'ajout, de suppression ou de modification.

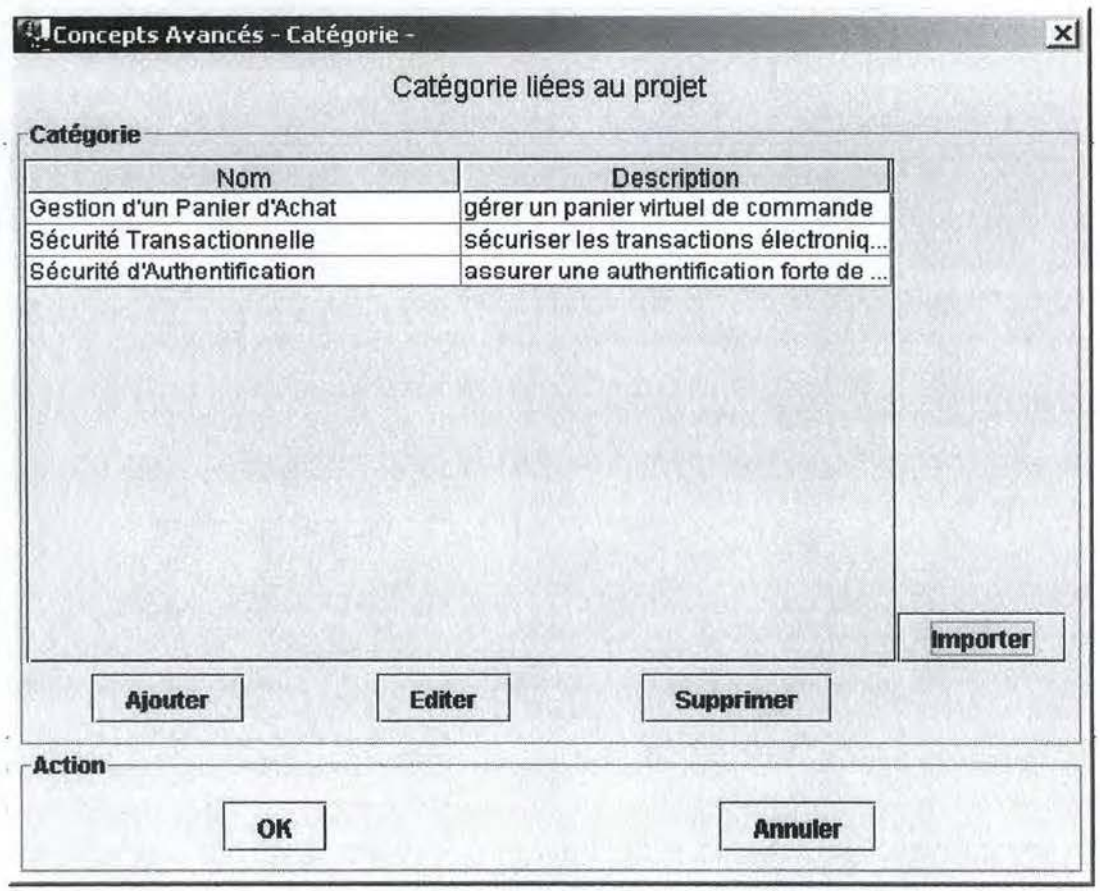


Figure 12 : IHM Gestion des concepts avancés

La Figure 13 ci-dessous est le diagramme d'activités représentant la gestion classique d'un concept avancé.

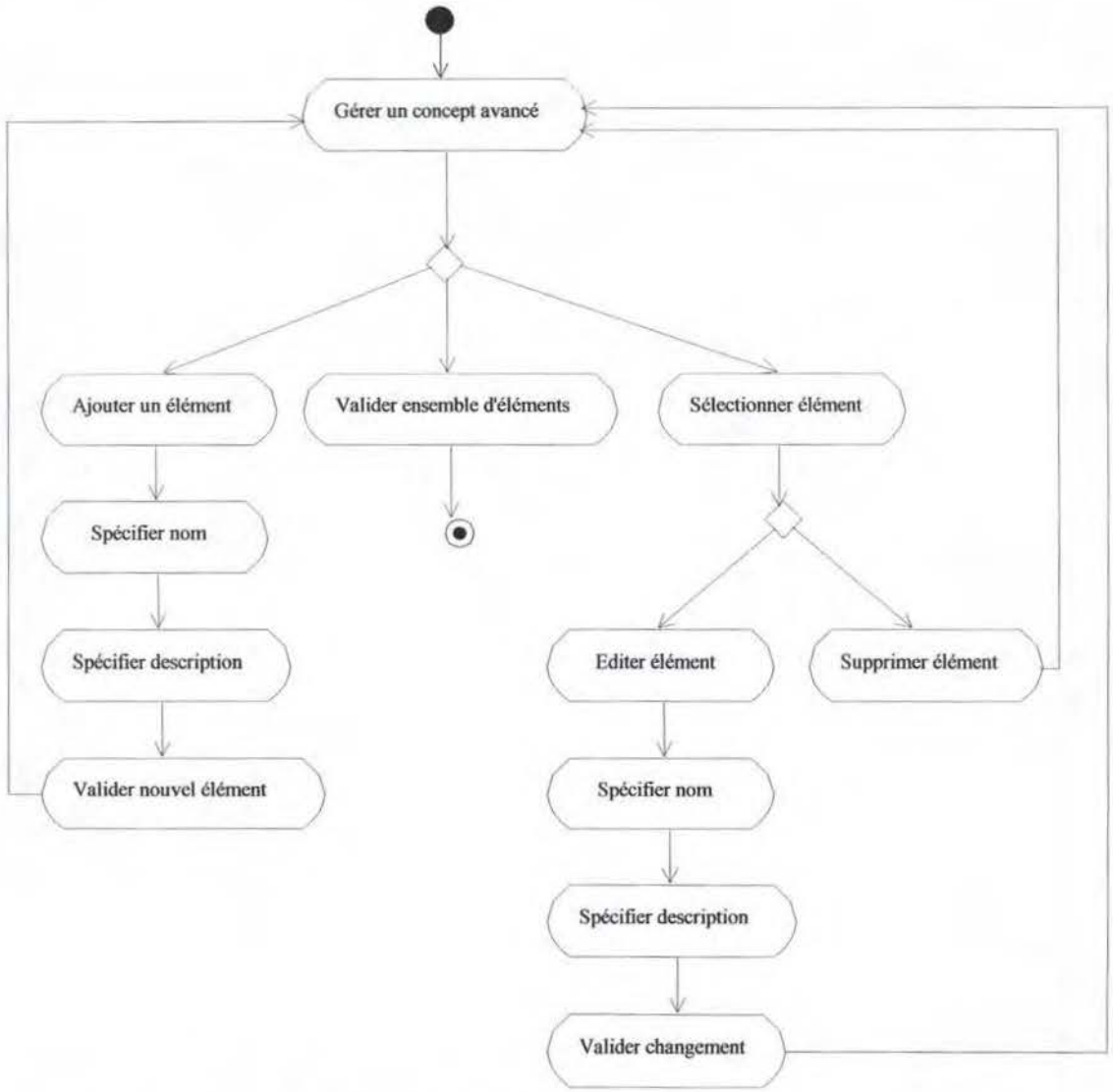


Figure 13 : Diagramme d'activités de la gestion classique des éléments des concepts avancés

- Personnaliser les préférences de projet

  Il est possible de modifier tous les éléments composant les préférences d'un projet.

  - **la terminologie**
  - **le système de pondération**
  - **le système de notation**
  - **le glossaire**

  Ces éléments peuvent être modifiés par importation depuis la base de données administrateur ou par des fonctionnalités d'ajout, de suppression, de modification.

- **Les structures de description** : Il est possible de définir la structure de description utilisée pour chaque partie du cahier des charges (présentation, exigences fonctionnelles, exigences non fonctionnelles, critères d'appel d'offres). Ces structures sont choisies parmi celles définies par l'administrateur.

- **L'interpondération des parties du cahier des charges** : L'utilisateur OPAL peut définir lui-même l'importance relative de chaque partie du cahier des charges par rapport aux autres.

Une fonctionnalité d'enregistrement permet de sauver l'ensemble de ces préférences de projet dans une base de données sous forme de modèle de préférences. Ces modèles peuvent être réutilisés lors de la création d'un nouveau projet.

3.2.1.4 Animer l'appel d'offres

L'animation de l'appel d'offres se décompose en fonctionnalités comme suit :

- Identifier les fournisseurs

Cette fonctionnalité permet l'obtention d'un ensemble de fournisseurs auxquels on souhaite soumettre l'appel d'offres. Cette liste est obtenue par importation depuis la base de données administrateur ou par ajout classique (après saisie des données relatives à ce fournisseur). Cette fonctionnalité n'est pas implémentée dans cette version d'OPAL.

- Gérer les lettres d'accompagnement

L'utilisateur peut rédiger une lettre d'accompagnement qui sera remise au fournisseur. Cette lettre est soit commune à tous les fournisseurs, soit personnalisée.

- Gérer le questionnaire d'appel d'offres

Chaque exigence peut être associée à une question. L'ensemble de ces questions forme le questionnaire qui sera soumis aux différents fournisseurs. Il est également possible d'ajouter un commentaire à chaque question afin d'être plus clair.

- Noter les exigences par fournisseur

Cette fonctionnalité permet de noter les réponses des fournisseurs aux questions qui leurs ont été posées. Cette notation se fait selon le système de notation du projet et constitue la base de l'analyse des offres.

- Analyser les offres

L'analyse des différentes offres permet de visualiser un tableau récapitulatif de l'ensemble des réponses des fournisseurs. Cette grille pourra être adaptée en spécifiant des critères de mise en avant tels que des notations trop faibles pour des exigences ayant une grande importance. D'autres outils seront à la disposition de l'utilisateur afin de mettre en avant la sensibilité du résultat selon certains critères (pondération des grandes parties, ect.). Ces fonctionnalités ne sont pas encore implémentées.

3.2.2 OPAL Admin

Les fonctionnalités suivantes (Figure 14) ne sont accessibles que par le module administrateur d'OPAL et sont restreintes aux utilisateurs ayant des droits suffisants (administrateurs et super administrateurs).

Le but principal du module administrateur de OPAL est de gérer la capitalisation haute c'est-à-dire l'ensemble des informations accessibles à tout utilisateur du programme. L'accès à ce module a été restreint aux seuls administrateurs afin d'assurer une certaine homogénéité des projets OPAL.

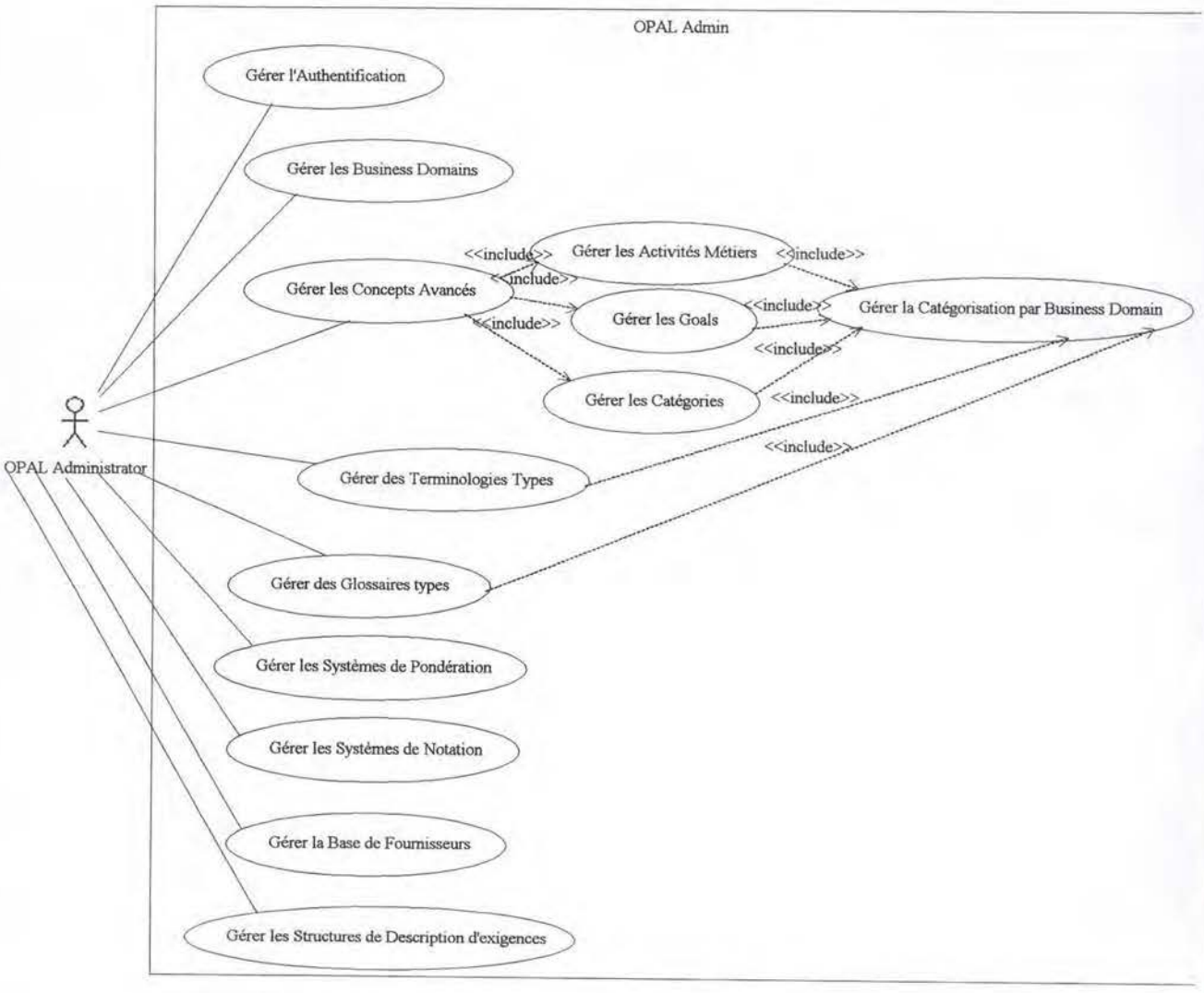


Figure 14 : Use case Diagram OPAL admin

3.2.2.1 Gérer l'authentification

Cette fonctionnalité permet de visualiser l'ensemble des utilisateurs OPAL ainsi que leurs droits d'accès. Elle permet également de créer un nouveau compte en saisissant les informations relatives au nouvel utilisateur (nom, prénom, droit d'accès, ...). Il est également possible de supprimer un compte ou bien de le modifier. Pour la création ou la modification d'un compte, seul un super administrateur peut donner des droits d'accès administrateur ou super administrateur.

3.2.2.2 Gérer les business domains

La gestion des 'business domains' a pour but l'obtention d'une liste de binômes (nom du domaine, description du domaine). Cette liste est obtenue via des fonctionnalités d'ajout, de modification et de suppression des 'business domain'.

3.2.2.3 Gérer la catégorisation par business

Cette gestion permet à l'administrateur de relier un certain nombre d'éléments (activités métiers, goals, catégories, terminologies types, glossaires types) à des 'business domains'. Cette relation a pour objectif de guider les choix du consultant OPAL en fonction du domaine dans lequel il se trouve.

3.2.2.4 Gérer les activités métiers

Cette fonctionnalité permet de définir un ensemble d'activités métiers ainsi qu'une définition et ce via l'ajout, la suppression ou la modification d'éléments. Celles-ci seront utilisées par le consultant OPAL afin de gérer la traçabilité des exigences. (Fonctionnalité non encore implémentée.).

3.2.2.5 Gérer les goals

Fonctionnalité similaire à celle de la gestion des activités métiers. (Fonctionnalité non encore implémentée).

3.2.2.6 Gérer les catégories

Fonctionnalité similaire à celle de la gestion des activités métiers. (Fonctionnalité implémentée).

3.2.2.7 Gérer les concepts avancés

Cette gestion est composée de la gestion des activités métiers, goals et catégories.

3.2.2.8 Gérer les terminologies types

Cette fonctionnalité permet à l'administrateur de créer un ensemble de terminologies types appelé 'terminology set'. La gestion de la terminologie permet de faire varier le vocabulaire utilisé dans l'IHM afin de correspondre au mieux avec les habitudes de travail des consultants OPAL. Chaque 'terminology set' se compose de plusieurs mots-clés représentant une notion particulière comme Cahier des charges, appel d'offres, ... Ces mots-clés doivent tous être associés à une traduction, c'est-à-dire au mot qui sera utilisé dans l'IHM de l'utilisateur ayant choisi ce 'terminology set'. Une définition peut également être mentionnée pour chaque mot-clé.

3.2.2.9 Gérer les glossaires types

Un glossaire est un ensemble de termes et de leurs définitions qui doivent être présents dans tout cahier des charges afin de mettre les différentes parties concernées d'accord sur la sémantique des termes utilisés dans celui-ci. La fonctionnalité 'Gestion des glossaires types' permet à l'administrateur de définir un ensemble termes-définitions qui pourront être directement utilisés par le consultant OPAL. Une fois encore, cet ensemble est obtenu par des fonctionnalités d'ajout, de suppression et de modification d'éléments.

3.2.2.10 Gérer les systèmes de pondération

Un système de pondération est un ensemble de trinômes labels, abréviations, points associés qui est utilisé pour pondérer les exigences. OPAL Admin permet de créer plusieurs systèmes de pondération qui pourront être utilisés par le consultant OPAL. Cette gestion est possible grâce à des fonctionnalités d'ajout, de suppression, de modification de systèmes de pondération et d'ajout, de suppression, de modification d'éléments de ces systèmes de pondération.

3.2.2.11 Gérer les systèmes de notation

Un système de notation est un ensemble de trinômes label, abréviation, notation utilisés pour noter la réponse d'un fournisseur à une exigence. Cette fonctionnalité est semblable à la gestion des systèmes de pondération.

3.2.2.12 Gérer la base de fournisseurs

Cette fonctionnalité permet le développement d'une base de fournisseurs. (Fonctionnalité non encore implémentée).

3.2.2.13 Gérer les structures de description d'exigences

Une structure de description d'exigences est la forme dans laquelle une exigence devra être exprimée. Dans OPAL Consulting, il est nécessaire de choisir quatre structures de description d'exigences (pour les parties présentation, exigences fonctionnelles, exigences non fonctionnelles et critère d'appel d'offres).
L'administrateur OPAL peut créer de nouvelles structures de description en leur affectant un type (générique, présentation, EXF, ENF, CAO) afin d'aider le consultant dans le choix des structures de description. La construction de la structure se fait par l'ajout, la modification, la suppression ou le déplacement de 'Parties' ou de 'Contenu'. Une partie représente une unité sémantique (par exemple un titre, etc.) dans laquelle il sera possible d'insérer un ou plusieurs 'contenu'. Ces contenus sont de plusieurs types (texte, textarea, images, listes non ordonnées, listes ordonnées) et sont dotés d'un attribut 'style' qui peut être soit texte plein, italique ou gras. Un outil de prévisualisation de la structure d'exigences en cours est également proposé à l'administrateur afin de faciliter son travail.

La Figure 15 montre l'interface permettant de gérer les structures de description d'exigences.

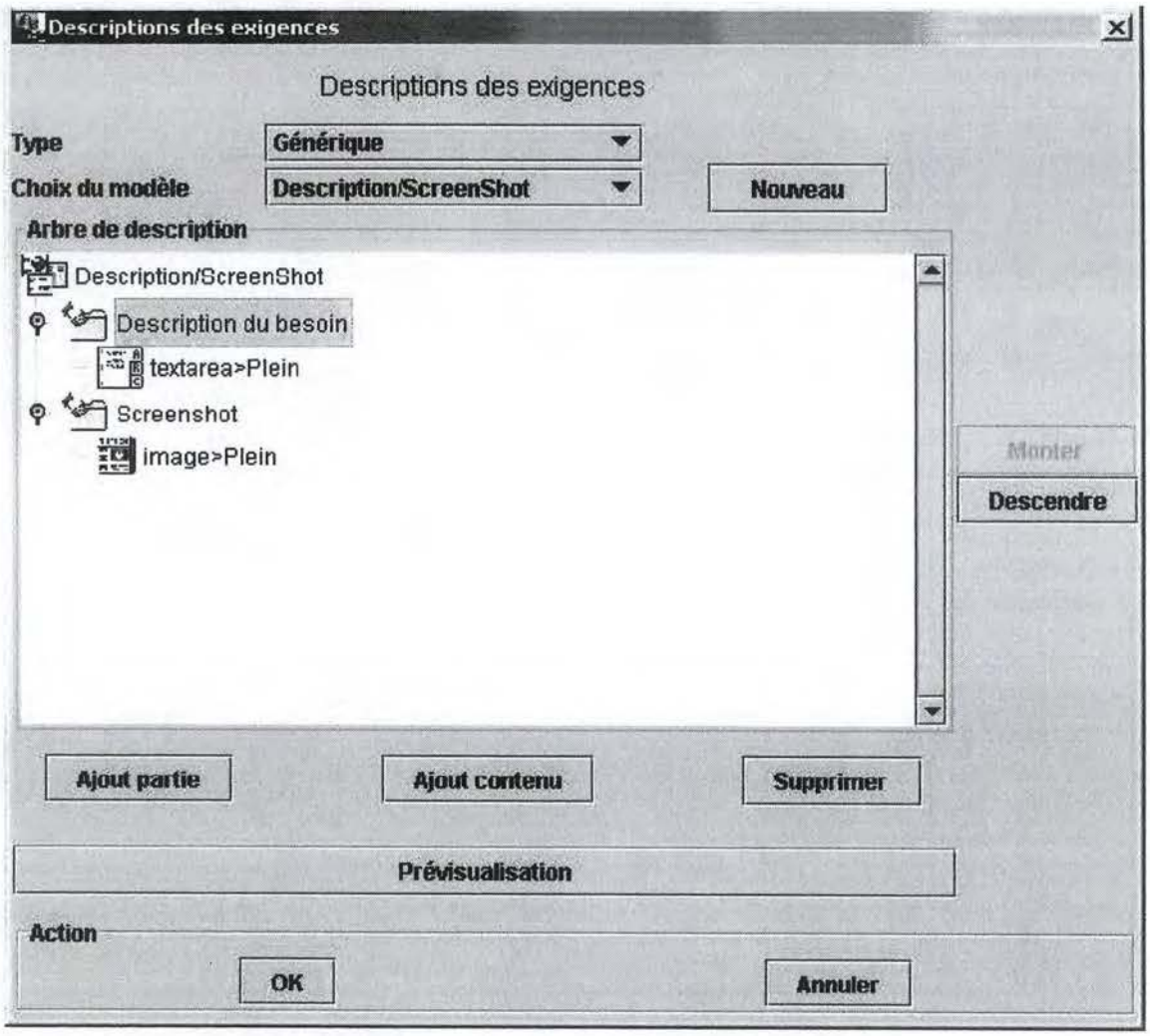


Figure 15 : IHM Gestion structures de description

La figure 16 est le diagramme d'activité représentant la gestion des structures de description. Ce diagramme représente donc les suites possibles d'actions nécessaires à la gestion des structures de description qui pourront être utilisées par le consultant.

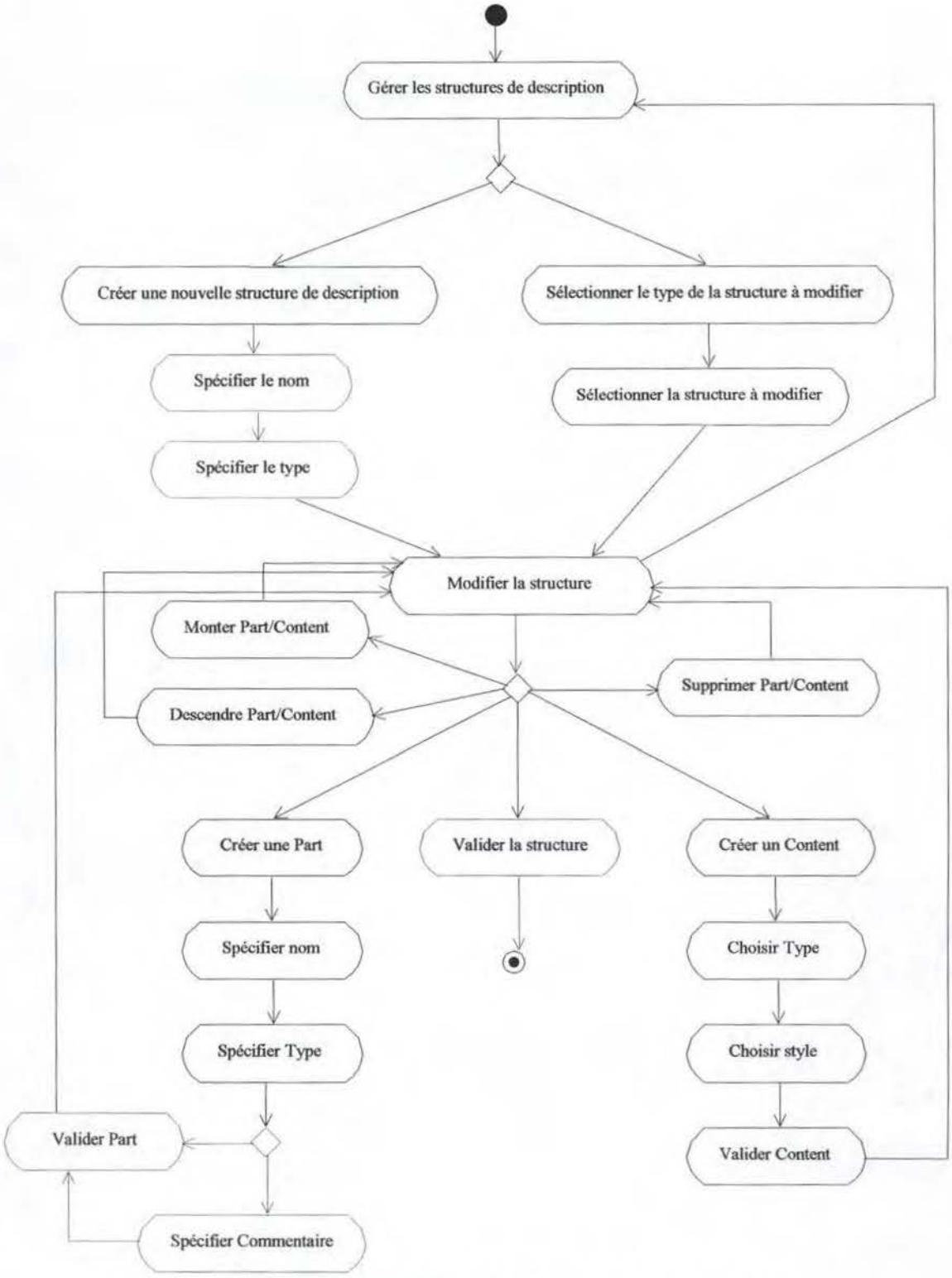


Figure 16 : Diagramme d'activités de la gestion des structures de description

3.3 Architecture

3.3.1 Architecture logicielle

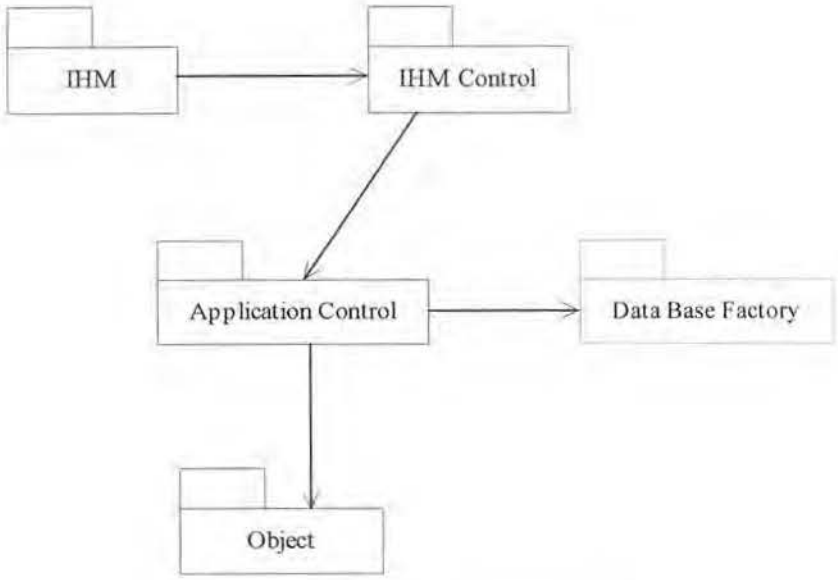


Figure 17 : OPAL architecture logicielle

Chacun des modules d'OPAL (Consulting et Admin) est divisé en cinq composants afin de regrouper tous les services d'un même type dans un package. Voici les principaux services fournis par ces composants :

- **IHM** : Le composant IHM regroupe toutes les interfaces du module. Dans ce composant, aucun traitement n'est effectué.
- **IHM Control** : Ce composant gère la navigation entre les différentes interfaces. Il permet également de relayer les demande d'informations ou de traitement du package IHM. Aucun traitement n'est effectué ici.
- **Application Control** : Regroupe toute la partie traitement du module. Ce composant relaye également les demandes d'informations (Accès à la base de données) ainsi que les opérations sur les Objets.
- **DataBase Factory** : Ce composant gère l'ensemble des opérations liées aux bases de données. Il s'occupe notamment de l'initialisation des connexions aux bases de données ainsi que des requêtes.
- **Object** : Ce package se charge de la représentation, sous une forme utilisable par les autres composants, des différents objets utilisés par un module. Il permet également d'effectuer des opérations sur ces objets (par exemple, créer un nouvel objet d'un type précis). Dans le module consulting, ce package permet la représentation du projet courant (exigences, concepts avancés,...).

3.3.2 Architecture technologique

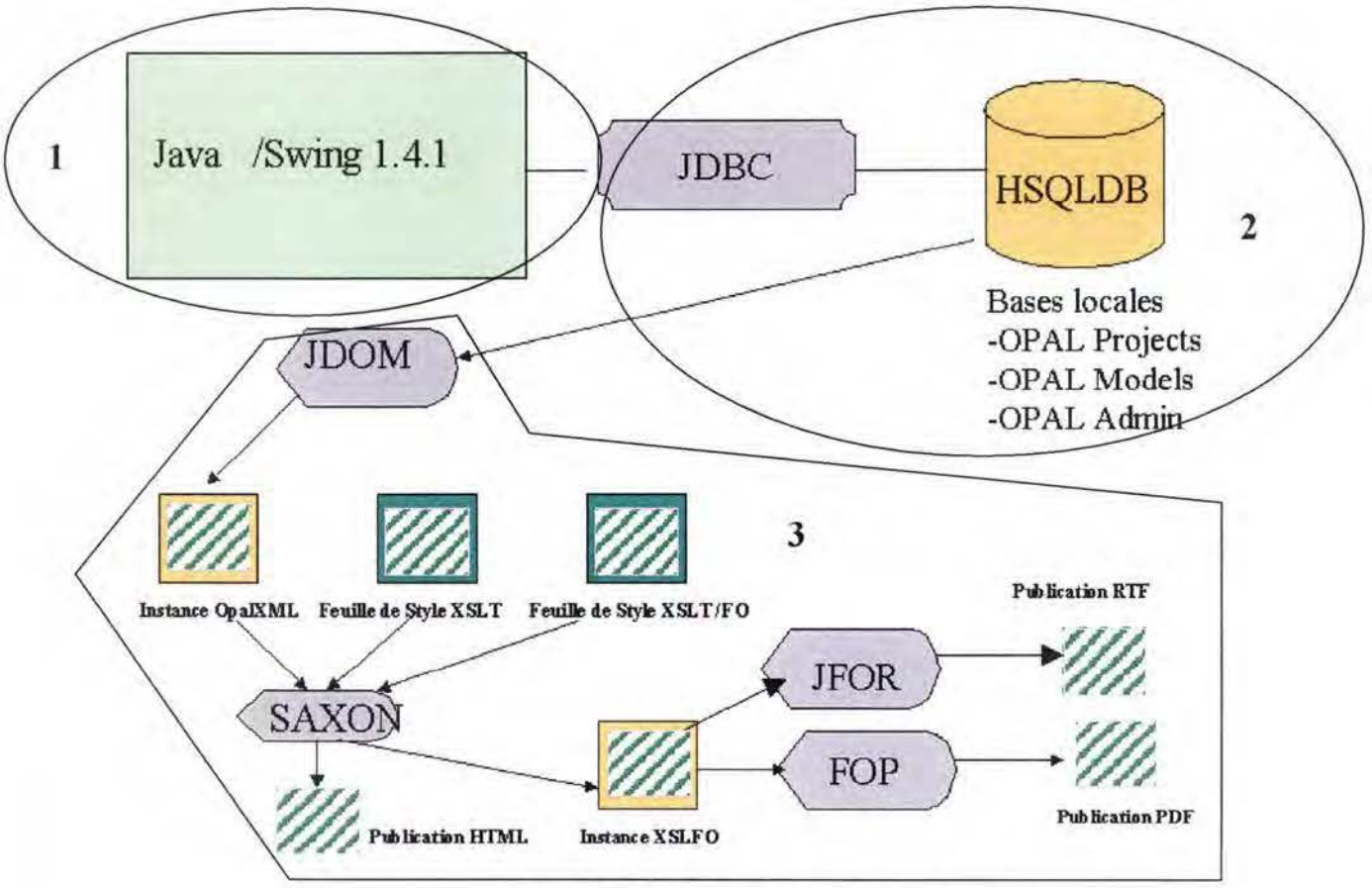


Figure 18 : OPAL technology model

1 : Le langage de programmation choisi pour OPAL est le JAVA[13]. Swing est une classe Java proposant des composants graphiques élaborés entièrement dessinés par Java, ce qui permet d'obtenir une interface graphique dont l'apparence n'est pas liée au système d'exploitation sur laquelle elle tourne. Toutes les interfaces de OPAL ont été uniquement développées à l'aide de swing.

2 : HSQLDB est le moteur de bases de données choisi pour le projet. Ce moteur (ainsi que l'API JDBC) permet au logiciel la sauvegarde de données. En effet, tous les projets, modèles de préférences et autres éléments ne sont stockés que sous forme d'entrée dans les bases de données. (Il n'est pas possible, à ce jour, de sauvegarder un projet sous forme de fichier)

3 : Ces composants permettent, via une structure XML, de publier les documents de projet sous format PDF ou RTF.

Les détails de tous les composants se trouvent à l'Annexe 1, de même qu'une brève description de la publication en PDF.

[13] www.java.sun.com

3.3.3 Architecture bases de données

Le modèle de données OPAL se décompose en trois schémas[14] de données bien spécifiques comme illustré par le schéma suivant :

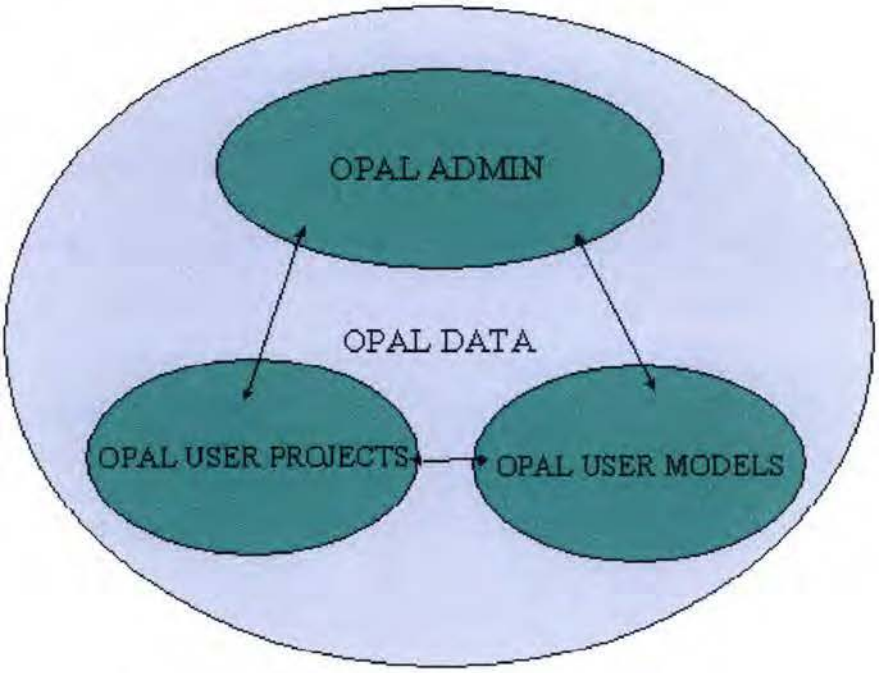


Figure 19 : OPAL Architecture bases de données

- **OPAL Admin** : Cette base de données contient tous les éléments jugés 'délicats' pour l'utilisateur. Ces données ne sont dès lors modifiables que par les administrateurs et supers administrateurs (sauf pour ce qui concerne les données personnelles relatives aux consultants car ceux-ci ont évidemment un droit de modification sur ces données).

- **OPAL User Models** : C'est à ce niveau que sont stockés les modèles de préférences créés par les utilisateurs de OPAL consulting.

- **OPAL User Projects** : C'est le schéma central de données et le cœur du logiciel OPAL. Il permet aux consultants de gérer l'ensemble des données relatives à leurs projets.

[14] L'Annexe 2 explique comment accéder aux schémas logiques des bases de données.

Conclusion

Le logiciel OPAL a donc atteint son but : Aider le consultant dans la création première du cahier des charges. Ce but est atteint par des fonctionnalités assez basiques mais permettant néanmoins une gestion assez fine des exigences.

OPAL ne remplace donc pas les autres logiciels du marché car ceux-ci proposent un panel de fonctionnalités beaucoup plus important. OPAL se limite en effet à la création du cahier des charges et à l'animation de l'appel d'offre mais ne propose aucun suivi du projet. Néanmoins, l'apport novateur de OPAL se situe au niveau de la réutilisation des cahiers des charges précédents.

Bien que le logiciel ne soit pas terminé (de nombreuses fonctionnalités prévues ne sont pas encore implémentées), une nouvelle version est actuellement en cours de spécification ce qui montre l'intérêt porté à ce logiciel. Il est donc probable que OPAL s'étoffera dans le futur et deviendra un logiciel de référence.

Conclusion générale

L'objectif principal de ce mémoire est de disposer de deux outils aidant à la construction de cahiers des charges. Le premier outil développé est un outil méthodologique basé sur un état de l'art en matière d'exigences non fonctionnelles. Le second est un outil logiciel d'aide à la création de cahiers des charges.

1. L'outil d'aide méthodologique

   L'outil d'aide méthodologique consiste, premièrement en un état de l'art en matière d'exigences non fonctionnelles. Ce document résulte de la mise en commun de deux normes et d'un template spécifiquement conçu pour la spécification des besoins lors du développement d'un logiciel. Les exigences non fonctionnelles retenues ont été ensuite modifiées si nécessaire afin d'être adaptées à l'acquisition de logiciels.

   De plus, les exigences ont été présentées sous forme d'exigences types, ce qui constitue l'apport novateur de cette partie. En effet, aucune norme, spécifique à l'acquisition logicielle, ne propose à l'heure actuelle un ensemble d'exigences non fonctionnelles pouvant être directement instanciées selon un contexte.

   La seconde partie de l'outil d'aide méthodologique est la définition de critères pour la sélection des exigences types. Ces critères constituent donc le deuxième apport de ce mémoire. En effet, cet outil permet au consultant d'avoir à sa disposition un support l'aidant à choisir, parmi un panel d'exigences non fonctionnelles, celles qui doivent figurer ou non dans le cahier des charges.

   De plus, la méthodologie proposée est accompagnée d'une illustration des instanciations possibles dans le domaine des ERP. Même si cet exemple peut paraître assez limité, il n'en reste pas moins très révélateur de la sélection semi-automatique des exigences types ainsi que de leurs instanciations lors de leur mise en contexte.

2. L'outil logiciel

   OPAL est un outil d'aide à la création de cahiers des charges. Il propose de nombreuses fonctionnalités qui permettent de créer un nouveau cahier des charges, de le construire mais également de gérer l'appel d'offre.

   La particularité de ce logiciel est la gestion avancée de la réutilisation des projets déjà créer. En effet, OPAL permet d'importer directement dans le projet courant un ensemble d'exigences appartenant à un cahier des charges déjà construit. Cette capitalisation permet donc un gain de temps important pour le consultant OPAL.

Une amélioration possible, dans le cadre de ce mémoire, aurait été de combiner les deux outils proposés. On pourrait, par exemple, imaginer un dialogue interactif entre OPAL et le consultant lors de la création d'un nouveau cahier des charges. OPAL demanderait les valeurs accordées aux différents critères utilisés dans l'aide méthodologique afin de pouvoir automatiquement générer les exigences non fonctionnelles du nouveau projet. Malheureusement, OPAL ne permet pas encore ce type de dialogue. Néanmoins, les outils proposés fournissent, malgré tout, une aide appréciable même s'ils ne sont pas encore interfacés.

Bibliographie

Template

[VOL98] Robertson J., Volere Requirements Process and Template, The Atlantic Systems Guild Ltd, 1998.

Normes

[IEE98] IEEE Std 830-1993, IEEE Recommended Pratice for Software Requirements Specifications.

[CC98] ISO/IEC 15408-1998, Critères communs pour l'évaluation de la sécurité des technologies de l'information Partie 2 : Exigences de sécurité fonctionnelles, version 2.0, Mai 1998.

Site internet

[ERP03] Diaz N., ERP ET QUALITE, http://perso.wanadoo.fr/nathalie.diaz/html/erpqualite.htm (Dernière mise à jours le 27/05/2003) (Date de consultation 15/08/2003).

Annexes

Annexe 1 : Détails des technologies employées dans OPAL

Java/Swing 1.4

Java est un langage de développement, produit par la société Sun et lancé le 23 mai 1995. Ce langage est orienté objet et comprend des éléments spécialement conçus pour la création d'applications multimédias. Swing est une classe Java proposant des composants graphiques élaborés entièrement dessinés par Java, ce qui permet d'obtenir une interface graphique dont l'apparence n'est pas liée au système d'exploitation sur laquelle elle tourne.

JDBC

JDBC (*Java DataBase Connectivity)* est un ensemble d'API permettant à un programme Java l'accès aux bases de données.

HSQLdb

hsqldb est un moteur de base de données relationnelles écrit en Java, avec un driver JDBC.

XML

La norme XML (*eXtensible Markup Language*) permet, avant tout, de stocker dans un fichier des informations structurées. On parle alors de document XML. Ce dernier est alors composé de textes libres et de balises (à la manière de ce que vous pouvez connaître avec l'HTML) possédant éventuellement des attributs. L'utilisation du XML permet entre autre:

- D'échanger des informations entre diverses applications.
- De générer des documents (HTML par exemple) ayant différents aspects selon l'utilisateur final.
- D'exporter/importer vers/de les bases de données.

  etc.

Les données sont indépendantes de l'affichage. Ainsi à partir d'un seul fichier XML, on pourra créer des documents sonores, pour téléphones portables, pour des appareils brailles, et bien sûr des pages HTML.

JDOM

Jdom permet de manipuler un document XML dans un programme Java. Il repose sur les mêmes principes que DOM mais plus simple car dédié au langage Java. Son but n'est pas de redéfinir un nouveau parseur mais de faciliter la manipulation au sens large de fichiers XML. Il permet en outre la lecture d'un document, la représentation sous forme d'arborescence, la manipulation de l'arbre, la définition d'un nouveau document, l'exportation vers plusieurs cibles. Jdom utilise SAX ou DOM pour parser le document XML.

XSLT

XSLT ((*eXtensible Stylesheet Language Transformation*)) est un langage sous-ensemble de XSL, qui permet d'effectuer des traitements et des transformations sur les données XML. Le document produit peut l'être dans différents langages (HTML, XHTML, WML,...).

XSL
XSL ((*eXtensible Stylesheet Language*)) est un langage dérivé de XML permettant la mise en forme par feuille de style des données XML.

Saxon
Saxon est un ensemble d'outils pour traiter les fichiers XML. Il comporte en particulier un processeur XSLT. Le processeur XSLT de Saxon permet de générer plusieurs documents à partir d'un seul fichier XML, de chaîner les feuilles de style à appliquer au document XML et de modifier les variables d'une feuille de style (qui ne peuvent normalement pas être modifiées, contrairement à ce que laisse supposer leur nom). Saxon est une alternative à Xalan que nous avons choisi car il semble plus performant pour le processing simultané d'instances XML.

FOP
FOP (*Formatting Object Processor*) est un processeur XSL (à ne pas confondre avec XSLT) acceptant en entrée un fichier XML comportant des Formatting Objects pour produire en sortie un document au format PDF.

JFOR
JFOR (*Java xsl-FO to Rtf converter*) est un formateur Java XSL-FO qui génère du RTF.

XSL/FO
XSL/FO (*eXtensible Stylesheet Language Formatting Option*) est une forme intermédiaire entre le média XML et le média de sortie. Vous alimentez le contenu de votre structure XML et la feuille de style XSLT avec processeur XSLT. Le résultat est XSL-FO. Vous alimentez cet XSL-FO, en ajoutant des mesures de polices de caractères et des éléments graphiques externes, dans un formateur XSL-FO. Le résultat est un document paginé (en PDF) qui peut être affiché ou imprimé.

Explication de la publication PDF

La publication s'effectue en transformant la représentation interne du projet courant en XML. Nous utilisons Jaxp pour l'analyse XML, cette API produit une instance XML. A partir de cette instance XML et d'une feuille de style XSLT, nous utilisons le processeur Xalan qui nous permettra de compiler le fichier XML. Xalan va nous produire une publication HTML et grâce à la feuille de style XSLT/FO, une instance XSLFO. Cette instance XSLFO est utilisée par le processeur FOP pour produire une publication PDF.

Annexe 2 : Accès aux schémas de bases de données

Les schémas des bases de données OPAL se trouvent dans le répertoire 'Schémas bases de données' du CDROM fourni.

Ces schémas ont été créés à l'aide de l'outil DBMAIN. Ce logiciel peut être obtenu dans sa version d'évaluation sur le site http://www.info.fundp.ac.be/~dbm/

Annexe 3 : Installation et utilisation d'OPAL

Vous trouverez dans cette annexe la procédure d'installation du logiciel OPAL ainsi qu'un bref scénario d'utilisation d'OPAL.

1 : Installation

Etape 1 : Installation JAVA

Installez JAVA version 1.4.1 minimum
Vous trouverez le fichier d'installation sur http://java.sun.com/j2se/1.4.1/download.html

Etape 2 : Copie fichiers

Copier le répertoire D:\opal\ sur votre disque dur. La variable %OPAL_PATH% utilisé par la suite représente l'emplacement des fichiers OPAL (par exemple c:\opal\)
Vous devez ensuite changer les droits des attributs des fichiers afin de retirer la lecture seule.

Etape 3 : Modification PATH

Modifier la variable d'environnement PATH en y ajoutant le répertoire BIN de l'emplacement ou java s'est installé. Par défaut 'C:\j2sdk1.4.0_02\bin'

Etape 4 : Création CLASSPATH

Créer une variable d'environnement appelé CLASSPATH et affecté lui les valeurs suivantes :

%OPAL_PATH%opalImpl;.; %OPAL_PATH%lib\hsqldb.jar ; %OPAL_PATH%lib\jaxp.jar; %OPAL_PATH%lib\jdom.jar; %OPAL_PATH% lib\saxon7.jar

Etape 5 : Compilation

Lancez une invite DOS et allez dans le répertoire : %OPAL_PATH%\opalImpl\opal\user\ihm\

Et exécutez javac *.java

Allez dans le répertoire %OPAL_PATH%\opalImpl\opal\admin\ihm\

Et exécutez javac *.java

Etape 6a : Lancement OPAL admin

Pour lancer OPAL ADMIN, placez vous dans %OPAL_PATH%\opalImpl\

Et exécuté 'java opal.admin.ihm.MainFrame'

Etape 6b : Lancement OPAL consulting

Pour lancer OPAL CONSULTING, placez vous dans %OPAL_PATH%\opalImpl\

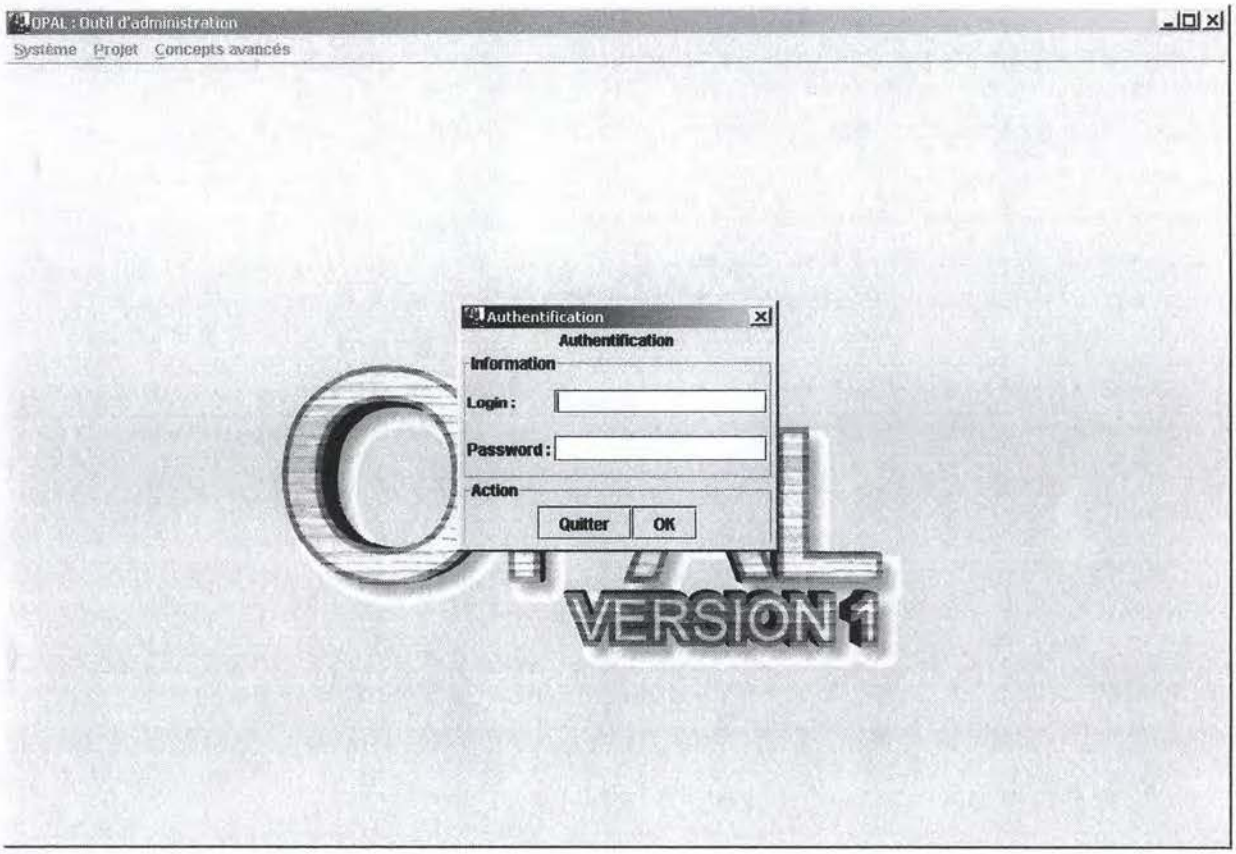
Et exécutez 'java opal.user.ihm.MainFrame'.

2 : Utilisation d'OPAL ADMIN

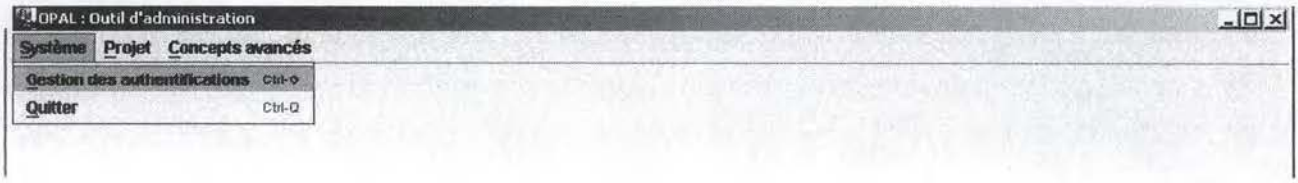
Le petit scénario proposé ci-dessous a pour but de vous familiariser avec l'IHM et l'utilisation du module administrateur d'OPAL. Il consistera simplement en l'ajout d'un nouvel utilisateur/administrateur puis en la création d'une nouvelle structure de description.

La première étape avant toute utilisation est de s'authentifier. Le super administrateur 'Lecteur' a été créé afin de vous permettre d'utiliser tous les modules d'OPAL :

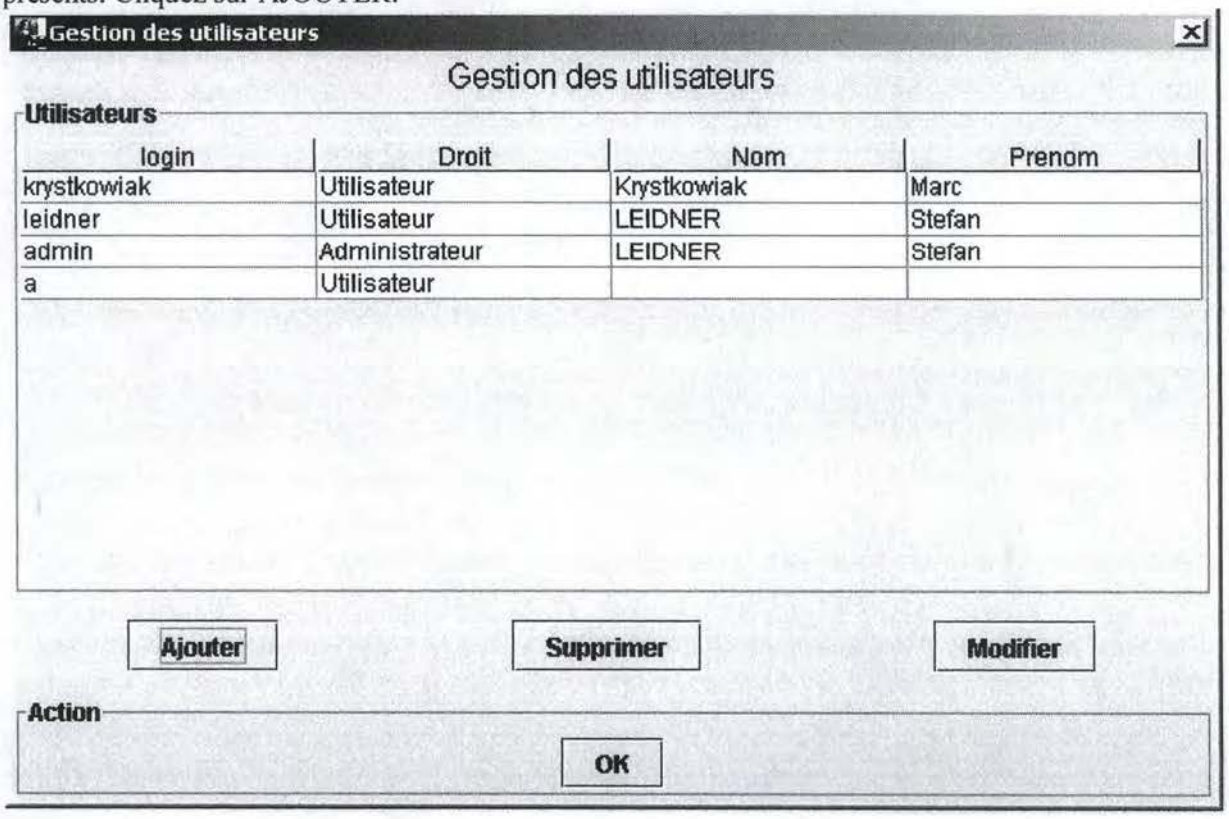
LOGIN : Lecteur
PASSWORD : Lecteur



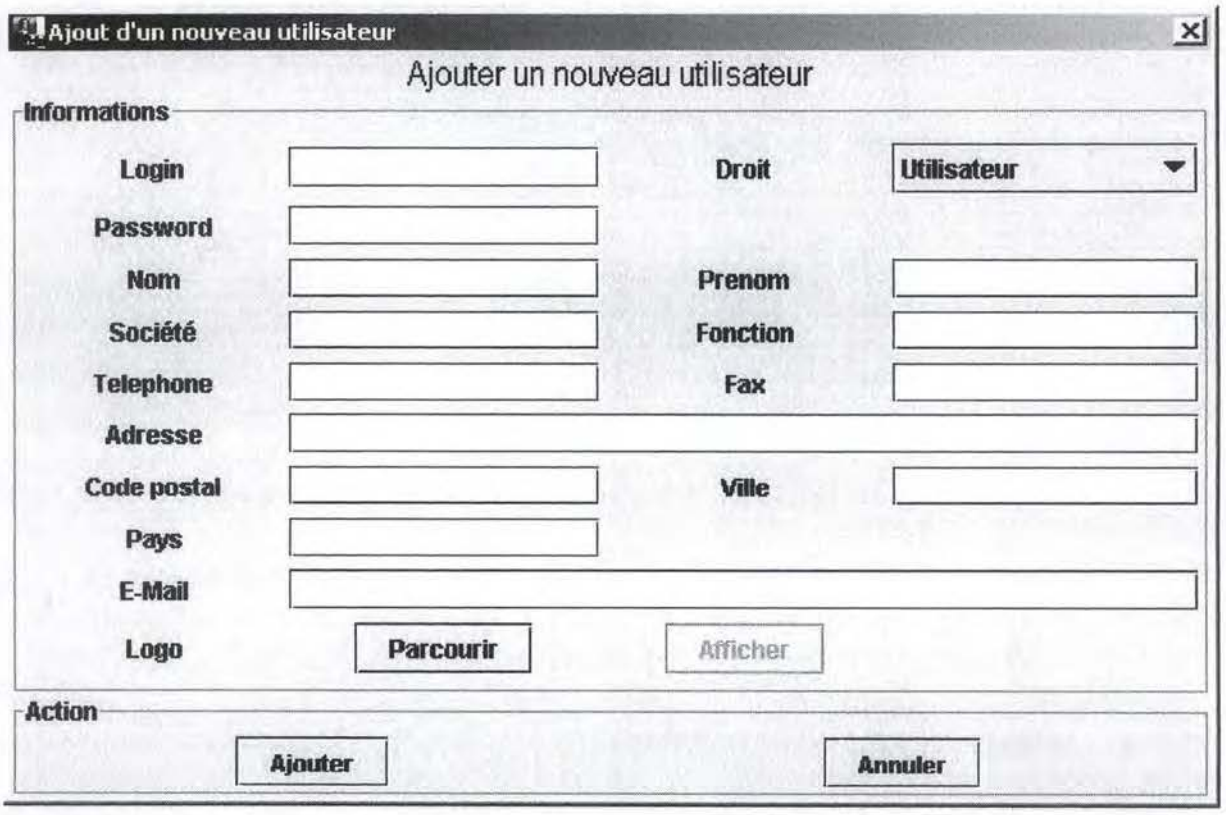
Une fois loggé, il ne reste plus qu'à créer un nouvel utilisateur. Pour ce faire, vous sélectionnez le menu SYSTEME puis GESTION DES AUTHENTIFICATIONS.



Vous vous trouvez maintenant devant la fenêtre permettant de gérer les utilisateurs. Cette fenêtre permet également de visualiser un certain nombre d'informations concernant les utilisateurs déjà présents. Cliquez sur AJOUTER.



La fenêtre qui apparaît vous permet d'introduire toutes les informations concernant le nouvel utilisateur. Entrez, par exemple, les informations vous concernant et sélectionnez également les droits d'accès super administrateur.

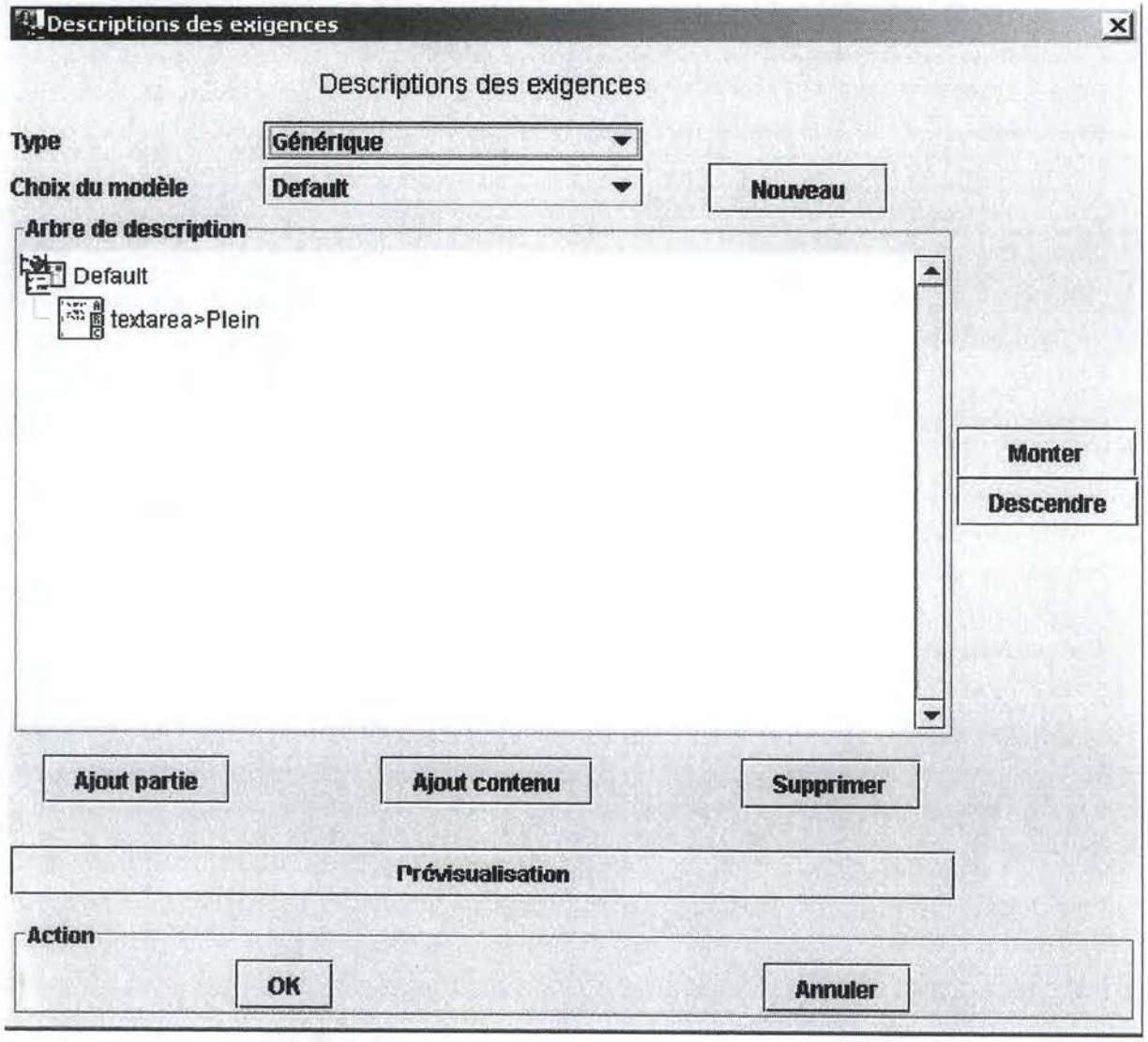


Une fois toutes les informations entrées, cliquez sur AJOUTER. Vous vous trouvez à nouveau sur la fenêtre récapitulative de l'ensemble des utilisateurs OPAL. Cliquez sur OK afin de valider les changements.
Votre nouvel utilisateur a donc été créé et vous pouvez dorénavant vous identifier à l'aide du login/password que vous avez spécifié.

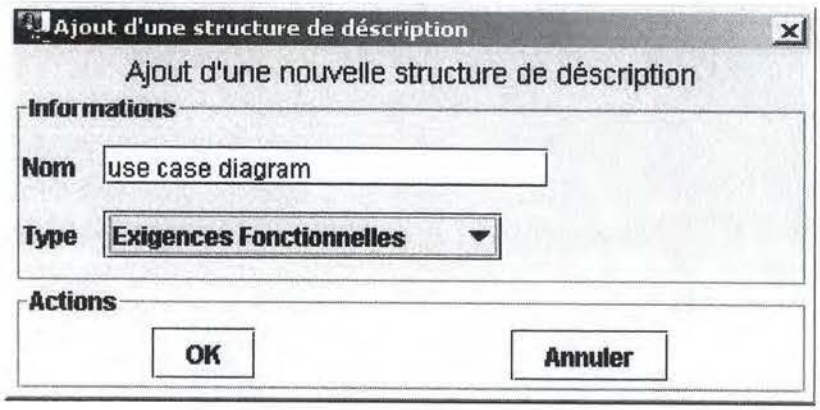
Nous allons maintenant créer une nouvelle structure de description afin de pouvoir décrire une exigence fonctionnelle à l'aide d'un use case. Ce use case ne sera composé que d'un champ contenant une image et d'un champ texte afin de permettre la description du use case diagram.

Sélectionnez le menu PROJET puis DESCRIPTIONS D'EXIGENCES.

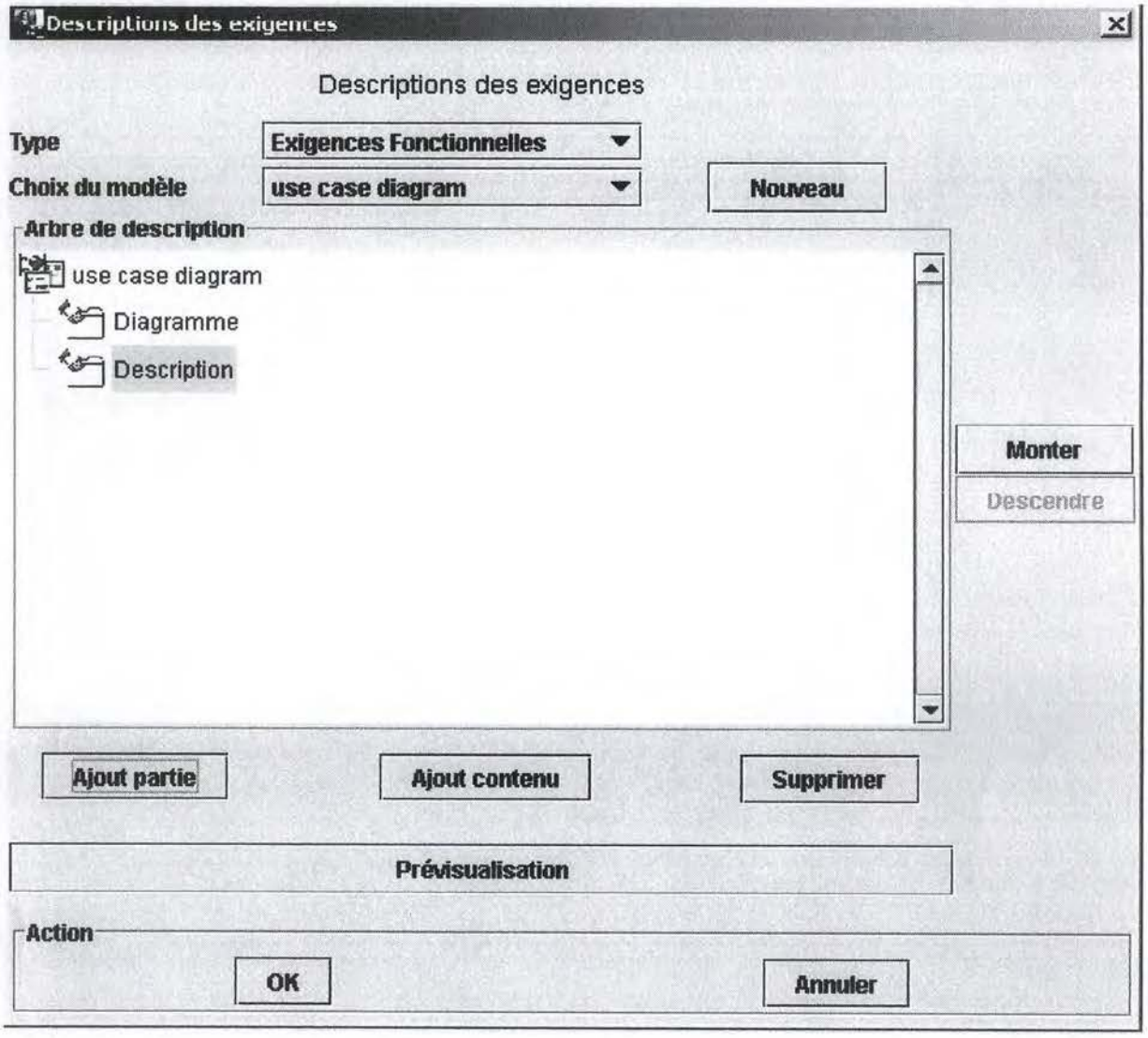
Vous vous trouvez maintenant dans l'outil de gestion des structures de descriptions d'exigences.



Cliquez ensuite sur NOUVEAU. Une fenêtre apparaît afin que vous puissiez choisir le type et le nom de la nouvelle structure. Entrez USE CASE DIAGRAM et sélectionnez le type EXIGENCES FONCTIONNELLES.

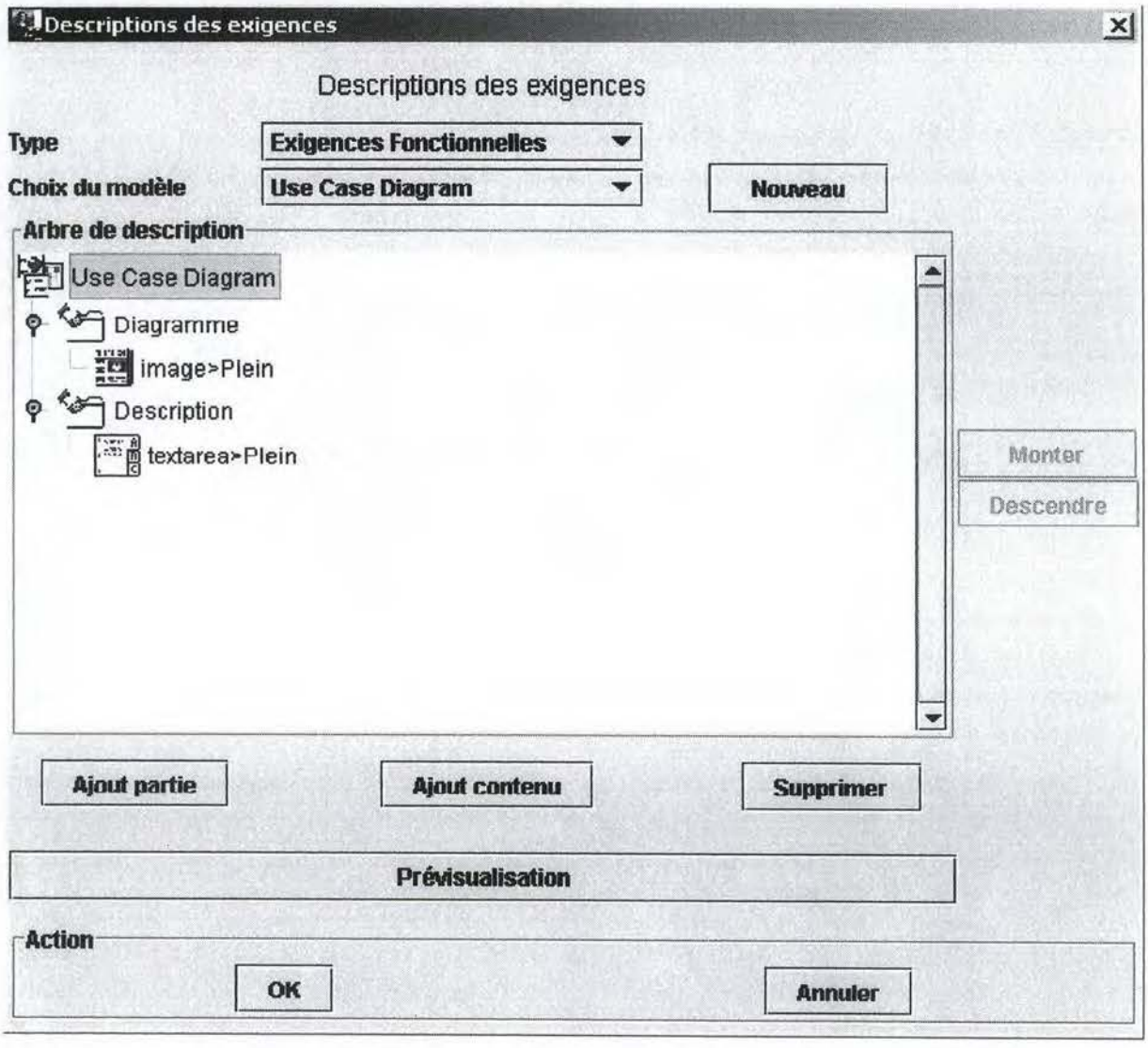


Vous arrivez sur une structure vierge. Sélectionnez le root de l'arbre (use case diagram) et ajoutez 2 parties : une nommée Diagramme et l'autre Description.



Ajoutez maintenant un contenu de type IMAGE dans la partie Diagramme. Pour ce faire, sélectionnez le nœud 'Diagramme', cliquez sur 'Ajout contenu', sélectionnez le type 'image' et validez. Utilisez la même technique pour ajouter un contenu de type 'Text area' au nœud Description.

Vous obtenez la structure suivante :



Il suffit ensuite de valider (OK) pour que le nouveau modèle de description 'Use Case Diagram' soit accessible aux utilisateurs de OPAL consulting.

3 : Utilisation d'OPAL CONSULTING

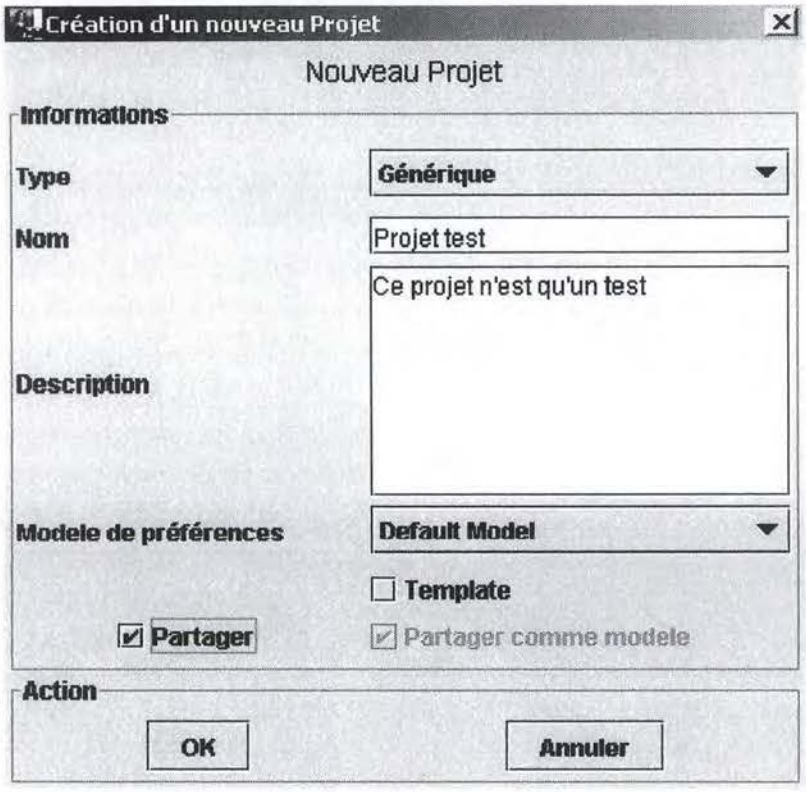
Ce bref scénario consistera à créer un nouveau projet et à importer certaines exigences d'un projet antérieur.

Lancez le module Consulting et entrez :

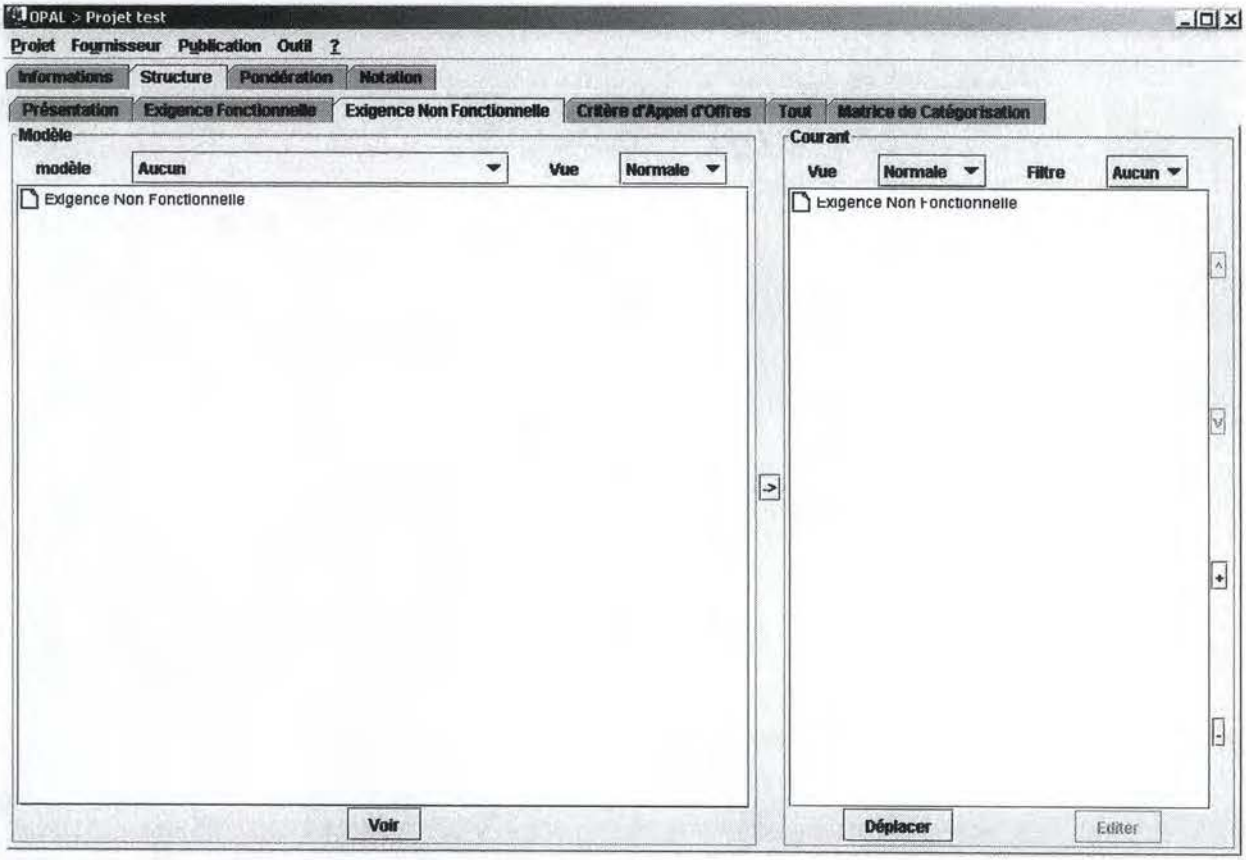
Login : Lecteur
Password : Lecteur

Ou le login/password de l'utilisateur que vous venez de créer dans le module administrateur.

Sélectionnez le menu 'Projet' et cliquez sur 'Nouveau'. Entrez les informations concernant le projet que vous désirez créer, par exemple :

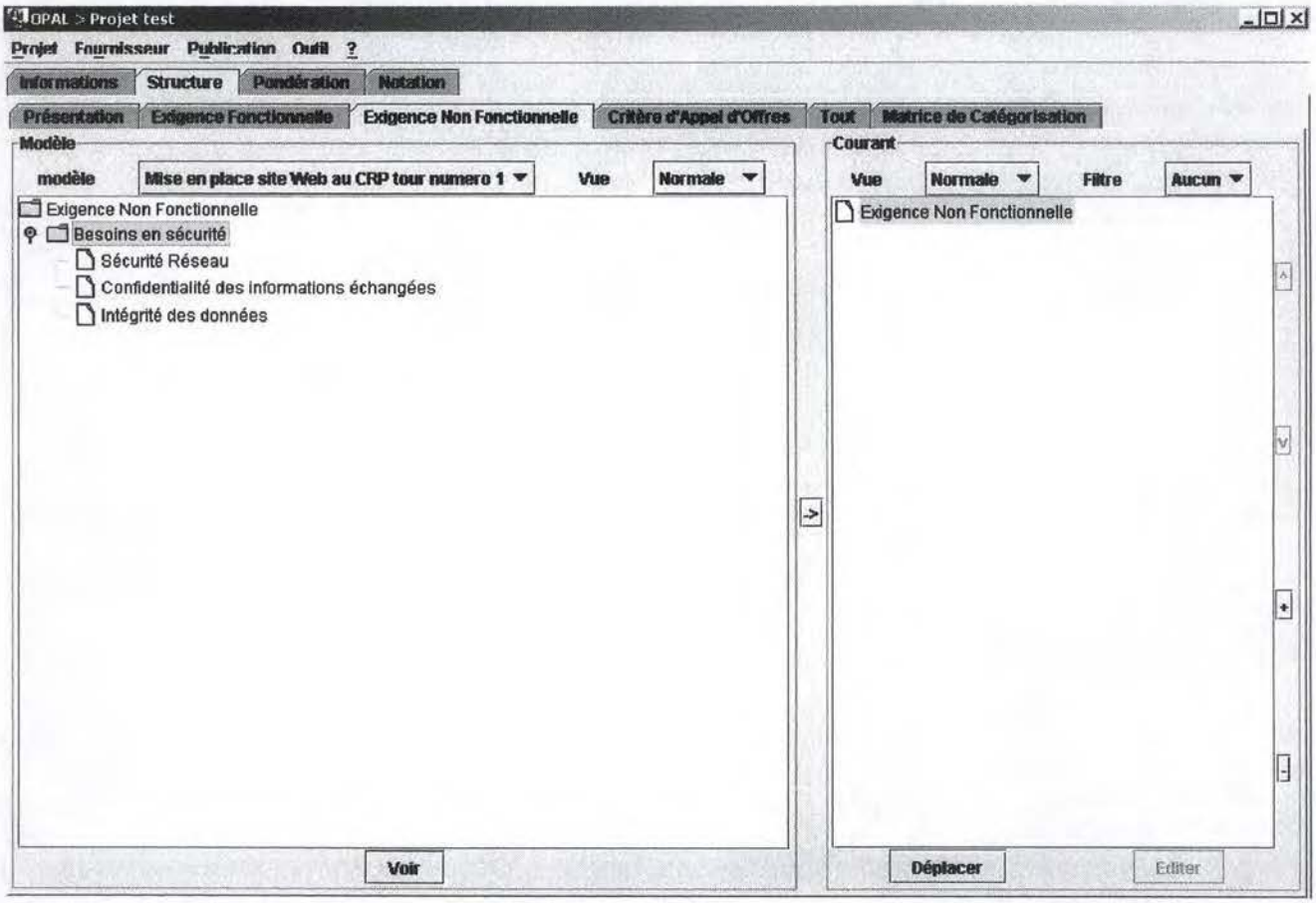


N'oubliez pas de définir un degré de partage du projet. Après avoir validé, vous vous trouvez devant la fenêtre principale de OPAL. Sélectionnez l'onglet 'Structure', ensuite le sous-onglet 'Exigences non fonctionnelles'.

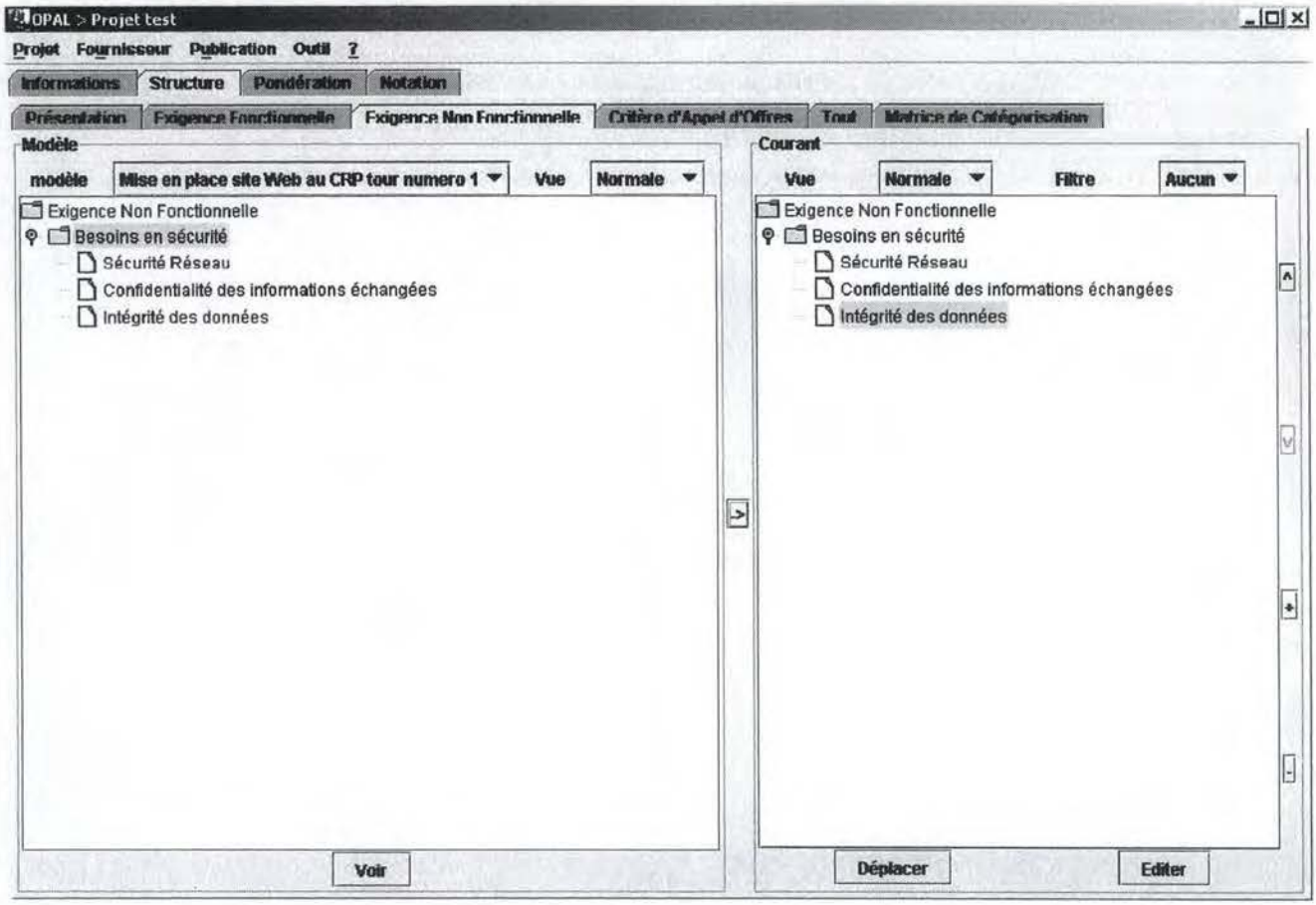


La partie gauche de l'écran représente le modèle d'où vous désirez importer des exigences. La partie droite représente le projet courant.

Dans la partie gauche de l'écran, choisissez le modèle 'Mise en place site Web au CRP tour numéro 1'. Puis, sélectionnez l'élément 'Besoins en sécurité'. Dans la partie droite, sélectionnez l'élément 'Exigences non fonctionnelles'.



Cliquez ensuite sur la flèche séparant les 2 parties de l'écran afin de transférer les exigences dans le projet courant.



Les exigences font maintenant partie du projet courant ce qui évidemment représente un grand gain de temps.

Vous pouvez maintenant modifier ces exigences, en ajouter d'autres,etc.